AFRL-IF-RS-TR-2006-102
Final Technical Report
March 2006

PREVENTING MISUSE OF OPERATOR PRIVILEGE (PMOP)

Massachusetts Institute of Technology

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. S474

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2006-102 has been reviewed and is approved for publication

APPROVED: /s/

ALAN J. AKINS
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** MARCH 2006	**3. REPORT TYPE AND DATES COVERED** Final Jul 04 – Jan 06

4. TITLE AND SUBTITLE
PREVENTING MISUSE OF OPERATOR PRIVILEGE (PMOP)

5. FUNDING NUMBERS
C - FA8750-04-C-0252
PE - 62301E
PR - S474
TA - SR
WU - SP

6. AUTHOR(S)
Robert Balzer, Howard Shrobe,
Neil Goldman and David Wile

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Prime:	Sub:
Massachusetts Institute of Technology CSAIL	Teknowledge, Incorporated
77 Massachusetts Ave 32-225	4640 Admiralty Way, Suite 1010
Cambridge Massachusetts 02139	Marina del Rey California 90292

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency	AFRL/IFGA
3701 North Fairfax Drive	525 Brooks Road
Arlington Virginia 22203-1714	Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-102

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Alan J. Akins/IFGA/(315) 330-1869/ Alan.Akins@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This document is the final report for PMOP, a project of the DARPA/IPTO Self-Regenerative Systems (SRS) program performed by MIT and Teknowledge.

Insiders are distinguished by the fact that they have been granted access to the system being defended, have been granted privileges on that system, and know how it operates. This means that traditional security mechanisms are ineffective against insiders.

PMOP assumes that the insider has all the access needed for an attack, and focuses on detecting malicious behavior. Detection is based on unique sensors that monitor application-level user actions and an analyzer of the application-level user history relative to a role-based model of expected behavior that identifies both the types of behavior expected in a situation and the means for assessing the appropriateness of the behavior observed. The analyzer detects both intentional and accidental actions that harm the system. A suspicious behavior detector differentiates the two by inferring user goals and identifying plans consistent with that behavior. A level of suspicion is established by the relative degree to which the user's actions fit the role-based plans to the exclusion of the attack plans. The effects of suspected insider attacks are contained to protect the system.

14. SUBJECT TERMS
Regenerative Systems, Cyber Defense, Intrusion Detection, Insider Threat, Role-Based Model, Operator Behavior Monitor

15. NUMBER OF PAGES
65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Abstract

Insiders are distinguished from other attackers by the fact that they have been granted access to the system being defended, have been granted privileges on that system, and know how it operates. This means that traditional security mechanisms are ineffective against insiders because they already have all the capabilities these defenses are protecting/monitoring.

In this project we assumed that the insider has all the access, privileges, and knowledge needed for an attack, and focused on detecting the malicious behavior required to mount that attack. This detection is based on a unique set of sensors that monitor application-level user actions and an advanced malicious behavior detector that analyzes the application-level user history relative to a role-based model of expected behavior. This model identifies both the types of behavior expected in a situation and the means for assessing the appropriateness of the particular behavior observed.

The analyzer detects both intentional and accidental actions that harm the system. A suspicious behavior detector differentiates the two by inferring user goals from the observed behavior and identifying a set of plans consistent with that behavior. A level of suspicion is established by the relative degree to which the user's actions fit the role-based plans to the exclusion of the attack plans. This level of suspicion triggers unique effectors that contain the effects of suspected insider attacks (in a dynamic process-level virtual machine) to protect the system while additional evidence is gathered; administrators can determine whether to authorize or quarantine the contained actions.

Table of Contents

1. Technical Summary 1
2. Approach 2
2.1. Protect Legacy Systems 3
2.2. Misuse Prevention Architecture 3
2.2.1. Operational System Model 4
2.2.2. Behavior Monitor (Sensors) 5
2.2.3. Impending Harm Detector 6
2.2.4. Malicious Behavior Detector 6
2.3. Comparison with Current Technology 7
2.4. Delivered Capabilities 8
3. OASIS: A Testbed Example 9
3.1. Testbed Requirements 9
3.2. OASIS Dem/Val: A Legacy System 9
3.3. OASIS Operational Scenario Description 9
3.4. Misuse Prevention System Infrastructure 12
4. Technical Result: Impending Harm Detection 13
4.1. Application Semantics: the Operational System Model 14
4.2. Rule Violations 15
4.3. Implementation Details 16
5. Technical Result: Successful Detection of Malicious Operator Attacks 16
5.1. Inferring Intent 17
5.2. Vulnerability Analysis 17
5.3. Expected Behavior Analysis 18
5.4. Plan Generation 18
5.5. Plan Recognition 18
5.6. Systematic Monitoring 19
5.7. An Example of Intent Analysis 19
6. Technical Result: Resource-Corruption Attacks Defended 21
7. Measurements of Success 23
7.1. Satisfying Self-Regenerative Systems (SRS) Program Objectives 23
7.1.1. Thwart or Delay at least 10% of Insider Attacks 23
7.1.2. Learn its own Vulnerabilities over time 23
7.1.3. Ameliorate those Vulnerabilities 24

7.1.4. Regenerate Service After Attack 24
7.2. Validation: Red Team Experiment 26
7.2.1. An Experiment to Determine Ability to Thwart Insider Attack 26
7.2.2. Red Team Experiment Results 26
7.3. Future Improvements 26
8. Conclusions: an Infrastructure for Self Generation 27
8.1. How can a plan be harmful? 27
8.2. What Are the Consequences of Harm? 28
8.3. Broadening the Vulnerability Analysis 29
8.4. Coordinated Attacks 30
8.5. Conclusion 30
9. Key Personnel 31
10. References 31
Appendix A: Jess Rules for Mission Data Corruption Detection 35
APPENDIX B: Initial Data 44

Table of Figures

Figure 1: Misuse Prevention Architecture 4
Figure 2: OASIS Dem/Val Scenario 11
Figure 3: The MAF/CAF GUI 12
Figure 4: Instrumented Dem/Val System Architecture 13
Figure 5: Thwarted Attempt to Write File 22
Figure 6: Red Team Configuration 25

Table of Tables

Table 1: OASIS Dem/Val Glossary of Terms 10

1. Technical Summary

Insiders are distinguished from other attackers by the fact that they have been granted access to the system being defended, have been granted privileges on that system, and know how it operates including the critical resources on which it depends. This means that traditional security mechanisms such as access/authorization controls or information gathering anomaly detectors are ineffective against insiders because they already have all the capabilities these defenses are protecting/monitoring (a few attacks may require the insider to acquire additional capabilities, but that just means that the traditional "outsider" security mechanisms can contribute for these cases).

In this project we assumed that the insider has all the access, privileges, and knowledge needed for an attack, and focused on detecting the malicious behavior required to mount that attack. This detection is based on a unique set of sensors that monitor application-level user actions and an advanced malicious behavior detector that analyzes the application-level user history relative to a role-based model of expected behavior. This model identifies both the types of behavior expected in a situation and the means for assessing the appropriateness of the particular behavior observed. The assessment uses a wide variety of mechanisms for determining the appropriateness of an action such as safety models, "plant" models, design rules, best practices, and heuristics.

The analyzer detects both intentional and accidental actions that harm the system. A suspicious behavior detector differentiates the two by inferring user goals from the observed behavior and identifying a set of plans consistent with that behavior. These plans are extracted from a library and include both plans associated with the user's role and attack plans (both generic and site specific).

A level of suspicion is established by the relative degree to which the user's actions fit the role-based plans to the exclusion of the attack plans. This level of suspicion triggers unique effectors that contain the effects of suspected insider attacks (in a dynamic process-level virtual machine) to protect the system while additional evidence is gathered; administrators can determine whether to authorize or quarantine the contained actions.

Two unique capabilities result from detecting attacks based on model-based predicted harm (about to be) caused to a system:

- There is no need to update the defense as new insider attacks are discovered or new ways to obfuscate them are invented.
- Attacks based on corrupted operand values or the situation in which operations are invoked can be detected and blocked.

Because our attack detectors are model-based, and thus harder to fool, the vast majority of insider attacks are detected and blocked.

2. Approach

The assumption that the insider has all the access, privileges, and knowledge needed for an attack – which defines what it means to be an insider – means that insider attack detection and thwarting must be based on the attack behavior itself. Although detection could occur after the fact, from observation of the damage caused, thwarting requires the detection to occur before the damage has been caused so that it can be prevented.

The detection must therefore be based on pre-damage activity – namely the user's commands or directives to some software system and that software's execution of system level operations to affect those commands or directives. Neither of these phenomena is normally available, but can be made so through proper instrumentation (discussed below).

Existing detection technology is pattern-based with behavior monitors recognizing known harmful patterns and anomaly detectors recognizing deviations from known acceptable patterns. We have augmented existing pattern-based technologies with a model-based approach that recognizes attacks by observing the effects of platform-level operations on a model of a system – those operations that cause damage or harm to a system are part of an attack on that system (whether by design or by accident).

This new detection technology has several advantages over the existing technologies. It is not limited to a known set of positive or negative instances and thus can detect novel attacks. It also is context or situation sensitive as the effect of an operation can depend on the state of the model as well as the parameters of the operation. This means that the detector can be much more selective than possible with pattern based detection. Moreover, it is tuned precisely to operations that harm the system rather than merely to adherence to, or avoidance of, positive or negative templates.

This opens up a whole new category of attacks (not just insider attacks) that can be detected and prevented – those that employ malicious values in otherwise benign operations. These malicious value data corruption attacks have remained problematical because existing security mechanisms just enforce type safety (i.e. the right type of value is used) to prevent execution faults resulting from the wrong type (including improperly formatted) of data being supplied. Without the type of operational system model we propose, there is no way to differentiate benign from malicious values within a type, especially when those determinations are context and/or state sensitive.

It should be noted that almost all insider attacks on financial systems, other than those meant to just crash the system or prevent its use, are examples of this currently unaddressed malicious-value data corruption class (e.g. putting the wrong amount into an account, depositing money in the wrong account). We surmise this is also true for insider attacks in general – though we know of no published data addressing this issue – because a tremendous amount of damage can be caused by merely telling the system to do the wrong thing (e.g. drop a bomb in the wrong location, deliver too little fuel, rendezvous at the wrong time).

2.1. Protect Legacy Systems

It would be nice if systems had built-in checks to prevent such misuse (whether from insider attack or operator error) but such protections tend currently to only occur in safety-critical systems (where they are usually required by law).

Rather than developing a technology for building software systems with such built-in checks we provide such “built-in” misuse checks for legacy systems by transparently encapsulating them in our Misuse Prevention architecture and screening operator commands and directives (by applying them to the system model) before passing them on to the encapsulated legacy system (see Figure 1). The encapsulated legacy system thus operates as if these misuse checks were built-in.

Our Misuse Prevention architecture can thus be applied to the huge inventory of critical legacy systems without requiring, or waiting for, them to be rebuilt according to some new set of standards. We believe this is a much more feasible transition strategy for our Misuse Prevention technology than any based on getting developers to incorporate our technology in their new products.

2.2. Misuse Prevention Architecture

Our Misuse Prevention Architecture (shown in Figure 1) encapsulates a legacy system to be protected so that operator actions are extracted from the legacy system by a *Behavior Monitor* and are screened for harmful effects before being passed to the legacy system, if found to be benign. This determination is made by applying the operator action to an *Operational System Model* to obtain a predicted state. This predicted state is then assessed to identify whether any damage or harm would have been caused to the legacy system in transitioning to this state by the *(Impending) Harm Detector*. If so, the operator action is blocked from the legacy system so that the predicted damage does not occur to it and the harmful operator action is passed to the *Malicious Behavior Detector* to determine whether the impending harm was accidental (i.e. operator error) or intentional (i.e. an attack by a malicious insider).

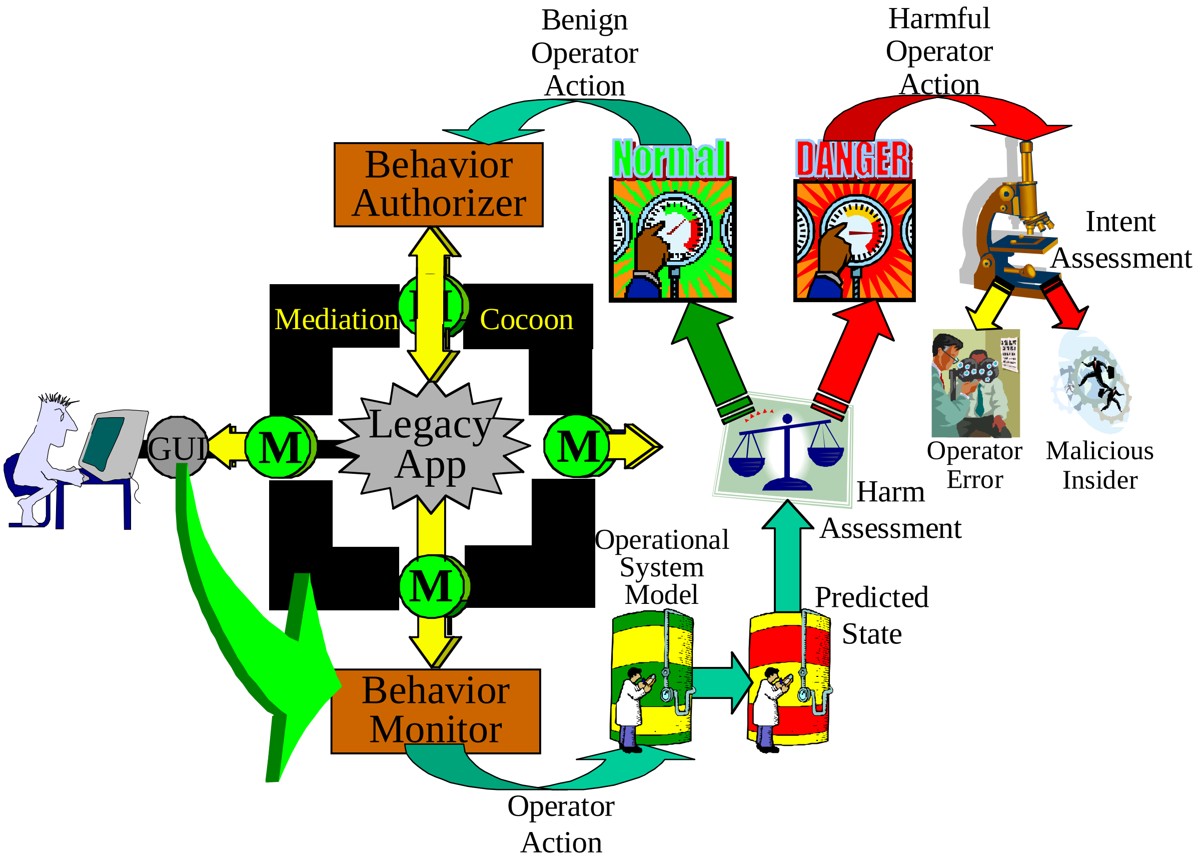


Figure 1: Misuse Prevention Architecture

2.2.1. Operational System Model

These additional capabilities do not come for free. They are based on the availability of an operational system model of the system, which must be constructed for each legacy system to be protected, and is only as good as the fidelity of that model. In the example system studied in this project the models were initially propositional rule bases, from which we inferred both the predicted state of the system and the likelihood of harm resulting from the change of state. In addition to modeling the system in its nominal state, one could also model the system in various states of compromise. For example, if a system has had its effective communication bandwidth reduced by a network denial of service attack, we need to infer the effect of user actions in that context, not simply the nominal context. In our earliest efforts the models were only able to detect harm when that harm was explicitly associated with the predicted state (e.g. detonating a missile before launching it) or it could be inferred from the transition into the predicted state on the basis of one of the following transition rules:

1. Actions that make resources unavailable to authorized users;

2. Actions that make resources available to unauthorized users;

3. Actions that inject disinformation into databases;

4. Actions that delete truthful information from databases.

More subtle attacks, as for example, injecting truthful but misleading information into a database, will have to wait for more refined operational system models and more powerful harm-inference reasoners.

2.2.2. Behavior Monitor (Sensors)

Detection is based on a unique set of application-level sensors that are able to detect user actions as they occur. In the past we demonstrated that instrumenting COTS products to build sensors that monitor the user's actions in real-time can be a highly successful approach to detection, as shown by our wrapper-based instrumentation of Microsoft Word and PowerPoint to provide application-level user histories. Here we leveraged our existing software and extensive experience with legacy applications protected by our Misuse Prevention Architecture.

Wrappers uniformly monitor in real-time the user's actions in a legacy application, including typing, editing, copying, pasting, and other actions. Actions are sensed uniformly whether they are invoked via menus, shortcuts, toolbars, or mouse-dragging. Relevant parameter values like the text that was entered, copied, or deleted are also accessible.

Conceptually, our wrapper is positioned between the application's high-level user interface and the application itself, allowing the wrapper to monitor the user's interaction with the application in terms of the application's own object model. This application model is defined by and accessed through a communication API that allows external code to query and manipulate this model (i.e. the application state).[1] These abstract communication interfaces are provided so that the application can be driven by a script in addition to an interactive user – thus enabling the application to be customized and automated (when users provide the script) and to be integrated and utilized by other applications (when the vendors of those products provide the script). The application's GUI conceptually utilizes this same interface to perform actions on behalf of the interactive user, though in practice it usually directly invokes the internal methods behind the interface.

The communication API and the application object model to which it provides access are the heart of our approach to providing an application-level behavior monitor. The API defines the application-level operations that the application can perform (whether invoked through the GUI or through script) and the operands needed for those operations.

[1] In our previous efforts, the communication interface was the Windows COM interface. For the present project, the application was entirely written in Java, so a Java wrapped version of the interface was used, as is described below.

2.2.3. Impending Harm Detector

By intercepting and mediating the calls from the GUI on the communication APIs, or the internal methods behind it, the user’s application-level actions can be captured and screened before those judged safe are allowed to pass onto the application itself for processing. By mediating the communication between the GUI and the application, the application-level actions are directly accessible (because they are the operations that the application is capable of performing). The GUI has performed the idiosyncratic translation from interface gestures (keyboard input, mouse clicks, and drags) to these application-level actions – we are merely mediating its communication with the application once this translation has occurred.

It should be noted that mediating this communication between the GUI and the application represented a departure from our previously existing technology to produce application-level user histories. That technology mediated the input to the GUI to identify when the user might be performing an action, allowed the (potential) action to proceed, and then determined whether a change had occurred – by comparing the application’s current state (as obtained through its COM API) with a cached version – and if so, determining what operation had been invoked by the nature of the changes between the cached and current states.

That approach was quite successful at inferring the application-level changes from the changes observed in the application’s state, but could not be used for this effort because here the application-level actions must be identified before they are acted upon by the application so that they can be screened for harmful effects with only the benign subset passed on to the application for processing.

This change from in-line, but after-the-fact monitoring instrumentation, to before-the-fact authorization instrumentation represented a major research objective.

2.2.4. Malicious Behavior Detector

We developed an advanced malicious behavior detector based on data received from these wrappers that analyzes the application-level user modification history relative to a role-based model of expected behavior. This model identifies both the types of behavior expected in a situation and the means for assessing the appropriateness of the particular behavior observed. The assessment uses a wide variety of mechanisms for determining the appropriateness of an action such as safety models, “plant” models, design rules, best practices, and heuristics. This analyzer detects both intentional and accidental actions that harm the system.

The suspicious behavior detector differentiates the two by inferring user goals from the observed harmful behavior, recent historical behaviors, and the set of plans consistent with the larger behavior context. These plans are extracted from a library and include both plans associated with the user’s role and attack plans (both generic and site specific). A level of suspicion is established by the relative degree to which the user’s actions fit the role-based plans to the exclusion of the attack plans.

It should be emphasized that attacks are detected *solely* on the basis of harm predicted by the Harm Detector; such harm would be caused by the pending application-level user command or directive, as inferred from the application of this pending command or directive to the operational system model. Because this attack detection is model-based there is no need to update this detector as new attacks are discovered or new obfuscation techniques are developed.

As explained below, we did develop a library of plans for both the role-based expected-behavior of users and the set of attacks to which it is vulnerable, but this library plays no part in the detection of an attack (based solely on predicted harm). Instead it is used after an attack has been detected to distinguish malicious intent (following an attack plan) from inadvertent operator error (following a role-based expected-behavior plan).

2.3. Comparison with Current Technology

Previous intrusion detection work has focused mainly on analyzing system level events (see, for example [Hofmeyr98]) in order to detect unauthorized access or modification of the system. This is partly because for an external attack, the system itself is what comes under attack, but it is also partly because that is the only level where sensor data has been widely available. Our approach instead used application level sensors, which are more appropriate for detection of insider attacks. User actions are monitored at the application level, much the way program actions are monitored at the system level in previous intrusion detection research. Information at this new level allows the detection of new attacks, in addition to allowing lower level detectors to consider what the user is doing.

Intrusion detection can be partitioned into two main categories: anomaly detection and Misuse Prevention. A great deal of research has been focused on anomaly detection as an intrusion detection technique, based on the assumption that even previously unanticipated attacks will cause the system to behave in a way that can be distinguished from normal behavior. A series of increasingly complex schemes have been investigated, ranging from simple n-gram based techniques [Forrest96] to Bayesian statistics [Anderson95, Porras97], data mining [Lee98, Barbara01], and neural nets [Ryan98]. While this research shows great promise in detecting anomalous application behavior and automatically learning how to do so from training data, all it can do is detect anomalous behavior. That anomalous behavior can be either benign and appropriate, or it can be malicious. Unfortunately, benign anomalous behavior is several orders of magnitude more common than malicious behavior, leading to high false positive rates for anomaly detection schemes. This is a fundamental limitation of the anomaly detection approach [Axelsson99]. Furthermore, there is no guarantee that all malicious behavior is anomalous; even at the system level certain attacks like race conditions are expected to be missed by anomaly detection [Forest96]. In fact, a clever attacker can craft his attack in such a way that a known anomaly detection system will not consider it anomalous [Wagner02].

Because benign anomalous behavior is several orders of magnitude more common than malicious behavior there appears to be little or no reason to front-end our model-based harm predictor with an anomaly detector – it would just reduce the set of user actions to check for producing harm. As those actions are produced at user speed, we do not see the performance improvement as being meaningful. Moreover, we would then be subject to any false-negatives produced by the anomaly detector – harmful actions not classified as anomalous. It appears to be far preferable to rely on our model-based harm predictor than on even the most sophisticated statistical approach.

Misuse Prevention attempts to detect the attacks themselves [Garvey91]. A wide variety of methods have been investigated, ranging from pattern matching [Kumar94] to rules based on state transitions [Ilgun95, Eckmann00]. These approaches need sensor data from the level where the attack is actually occurring in order to reliably distinguish between attacks and normal behavior. Furthermore, though we can build upon what has been learned previously, because explicit models of the attacks to be detected are required in existing Misuse Prevention schemes, models from previous research cannot simply be applied to the new insider attacks we are considering, nor can system level models be directly applied to application level data.

In addition to attack plans, our scheme also considers the effect of the intended actions on the system using a model of the system itself, and is able to consider the full context of the action to determine whether the action will cause harm to the system. The validity of the user's actions is also evaluated in the context of expected behavior based on the user's role and function within the organization. So in addition to detecting attacks that can be easily recognized as an instance of an attack from the attack library, our system will also be able to detect arbitrary deviations from normal behavior that lead to harm to the system. Such actions are dangerous regardless of whether they are malicious or not, and can safely be prevented whether they are malicious attacks or benign mistakes.

2.4. Delivered Capabilities

The following are the primary products of the project:

Operational System Model: an operational system model for a legacy application – primarily constructed from propositional rules – from which both the predicted state of the system, and the likelihood of harm resulting from the change of state can be predicted.

Misuse Prevention Architecture: a generic architecture for monitoring operator behavior in legacy systems at the level of application-specific commands or directives invoked by the operator, for matching that behavior against role-based plans, for modeling the effect of those commands or directives on the state of the legacy system, for assessing the benefit or harm of those effects, and for matching those effects and assessments against a set of insider attacks.

Operator Behavior Monitor: a component that mediates the communication between a legacy system's GUI and the system itself to extract the application level commands or directives initiated by the user/operator through that GUI so that they can be screened for harmful effects before being processed by the legacy system.

Matching Operator Behavior against Role-Based Plans: a component that compares operator behavior traces to behavior traces from operator and attack plans.

Malicious Behavior Detector: a suspicious behavior detector that differentiates between accidents and malicious behavior by inferring user goals from the observed harmful behavior, recent historical behaviors, and the set of plans consistent with the larger behavior context.

3. OASIS: A Testbed Example

3.1. Testbed Requirements

Our approach was predicated on applying operator commands and directives to an operational system model to predict the effect of those actions on the system, and when damage or harm would result, to determine whether the offending commands and directives were malicious or inadvertent operator error. We thus needed a real system with real users (so we did not need to build an artificial traffic generator), for which we could easily build an operational model, and an easily instrumented GUI.

Because we intended to work with a real system, we expected to need a failsafe way to disconnect the legacy system's GUI when malicious inputs are fed to the GUI during an insider attack, so that even if the malicious command or directive escapes detection, it is not passed to the legacy system.[2]

3.2. OASIS Dem/Val: A Legacy System

To demonstrate the applicability of our Misuse Prevention architecture to legacy systems, we chose a moderately large example legacy system to model and defend against insider attacks, the OASIS Dem/Val system developed under an earlier DARPA program. This system relies on the Joint Battlespace Infosphere (JBI) repository and communication protocol for coordinating and managing information from a variety of agents cooperating in the development of major military plans. The OASIS Dem/Val system developed air tasking orders for air cargo and munitions delivery and deployment and was created to demonstrate how existing military systems could interoperate with new components through the JBI infrastructure.

3.3. OASIS Operational Scenario Description

The motivation and actual scenario chosen to demonstrate our technology using the OASIS system is best described by the operations manual itself [Holzhauer 04, p. 4]:

As context for the OASIS scenario is Operation Allied Force, the NATO military operation fought primarily with air power and used to compel Serbia to cease hostilities against ethnic Albanians in Kosovo. This air operation allowed peace-keeping forces, on

[2] In our testbed system, about to be described, the system was never connected to a live response system, so this requirement was satisfied by default.

the ground, to carry out their mission to a successful conclusion. Much of the success in Kosovo was due to incredible efforts of the individuals involved in the planning and execution of operations, but a tool like the OASIS JBI would hopefully make their job easier.

Within this theater of operations, our scenario will be scoped to focus only on some of the functions performed by an AOC/TACC and its constituent planning cells. We will describe the separate planning processes that occur within the AOC/TACC in developing, refining, and executing an Air Battle Plan against WMD facilities, taking environmental factors such as weather and chemical plume hazards into consideration.

The associated use case models targeting and mission planning for air strikes against weapons of mass destruction (WMD) facilities. Weather and chemical plume/aerosol effects are taken into consideration during this mission. Weather changes affect predicted WMD plume dispersion, requiring the in-flight sortie to stand down in order to prevent undesirable propagation of the plume.

AOC – Air Operations Center
AODB – Air Operations Database
ATO – Air Tasking Order
CAF – Combat Air Forces
EDC – Environmental Data Cube
HTML – Hyper Text Markup Language
IO – Information Object
IR - InfraRed
JBI – Joint Battlespace Infosphere
JEES – Joint Environment Exploitation Segment
JWIS – Joint Weather Impact System
MAF – Mobility Air Forces
METAR – Meteorological Air Report
MIDB – Modernized Integrated Database
NATO – North Atlantic Treaty Organization
OASIS – Organically –Assured and Survivable Information Systems
SPI – Sensor Performance Impact
TACC – Tanker Airlift Control Center
TAF – Terminal Aerodome Forecast
TAP – Theater Air Planner
TBMCS – Theater Battle Management/Core Systems
TNL – Target Nomination List
TWS – Theater Weather Server
USMTF – US Message Text Format
WMD – Weapons of Mass Destruction
XML – extensible Markup Language

Table 1: OASIS Dem/Val Glossary of Terms

The process begins with the Targeting Cell in the Air Operations Center producing a Target Nomination List (TNL). The Combat Plans Cell then takes this TNL and builds the Air Tasking Order (ATO) assigning strike packages against each target. The ATO is then sent out to each unit/squadron who will be participating in the strikes. A few hours after the TNL is distributed, the Combat Operations Cell comes online to start monitoring weather conditions, readying aircraft, and implementing the ATO when it is built.

Throughout this planning process there is opportunity to take weather effects into account. Weather affects such things as weapon/sensor head selection, route selection, and attack timing. In addition, when considering an attack against a WMD facility, we have to assess where the released chemical materials will travel and ensure that neither non-combatants nor friendly forces will be harmed.

While the ATO is being built, the TACC in St. Louis is also planning an in-theater mission. An airlift mission to Prince Sultan Air Base is built that involves in-air refueling in the North Sea and landings in Aviano and Sigonella. This flight will be reconciled with the in-theater Director of Mobility (DirMob) to ensure that weather conditions and other factors will permit the flight. Once the DirMob approves the mission, notice is sent back to the TACC MAF planner for execution.

The ATO is then finalized and distributed to Combat Operations, who is responsible for mission execution and monitoring. After planes have departed for their targets, updated weather conditions become available from Air Force Weather Agency (AFWA.) After analysis of this latest weather by the Chemical Hazard Cell, it is predicted that a change in winds around the WMD site will cause a toxic plume to encroach a heavily populated area of non-combatants. The new plume pattern is passed to Combat Operations, who redirects the WMD sortie to stand down and return to base. Table 1 should help to clear up the meanings of most of the acronyms.

The normal flow of publications through the system is shown in Figure 2. A solid arrow indicates publication by the agent at the root of the arrow and the dotted arrows represent receipt of the information by the agent at the tip of the arrow (no matter which direction the arrow is pointing). Time progresses downward, although some of the timings are artificial. For example, it does not matter whether WLC is published before or after WH, because their consumers are disjoint and neither produces a publication before both are received.

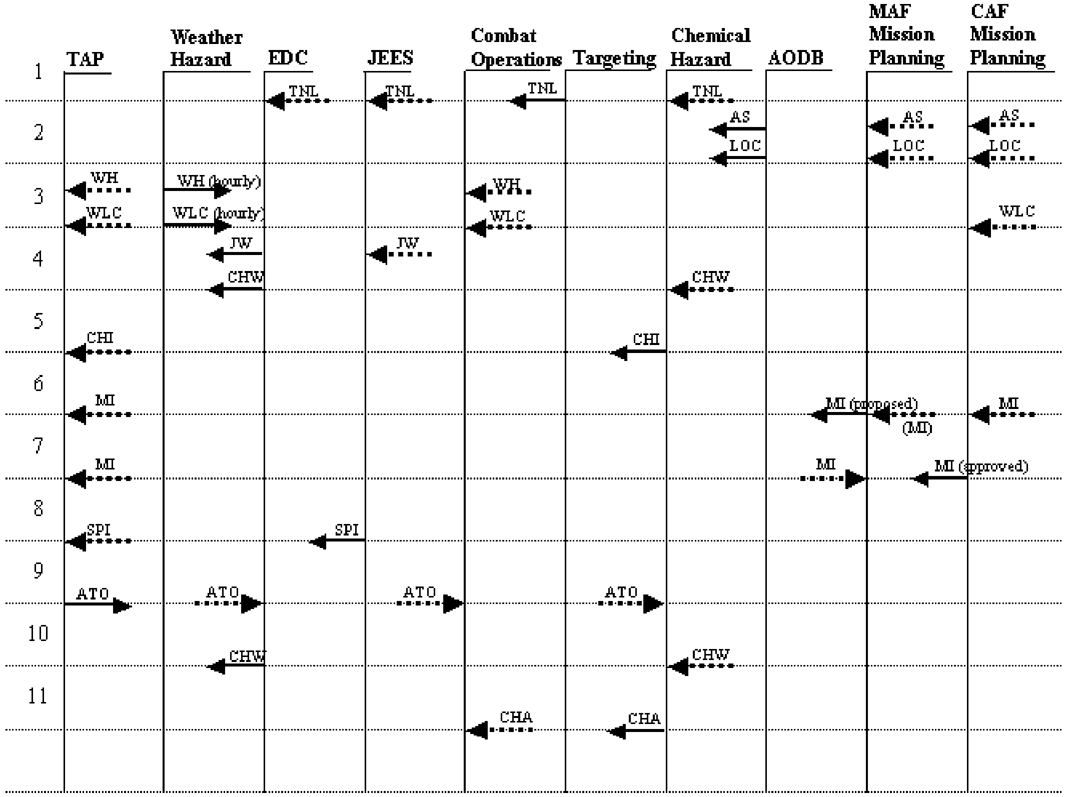
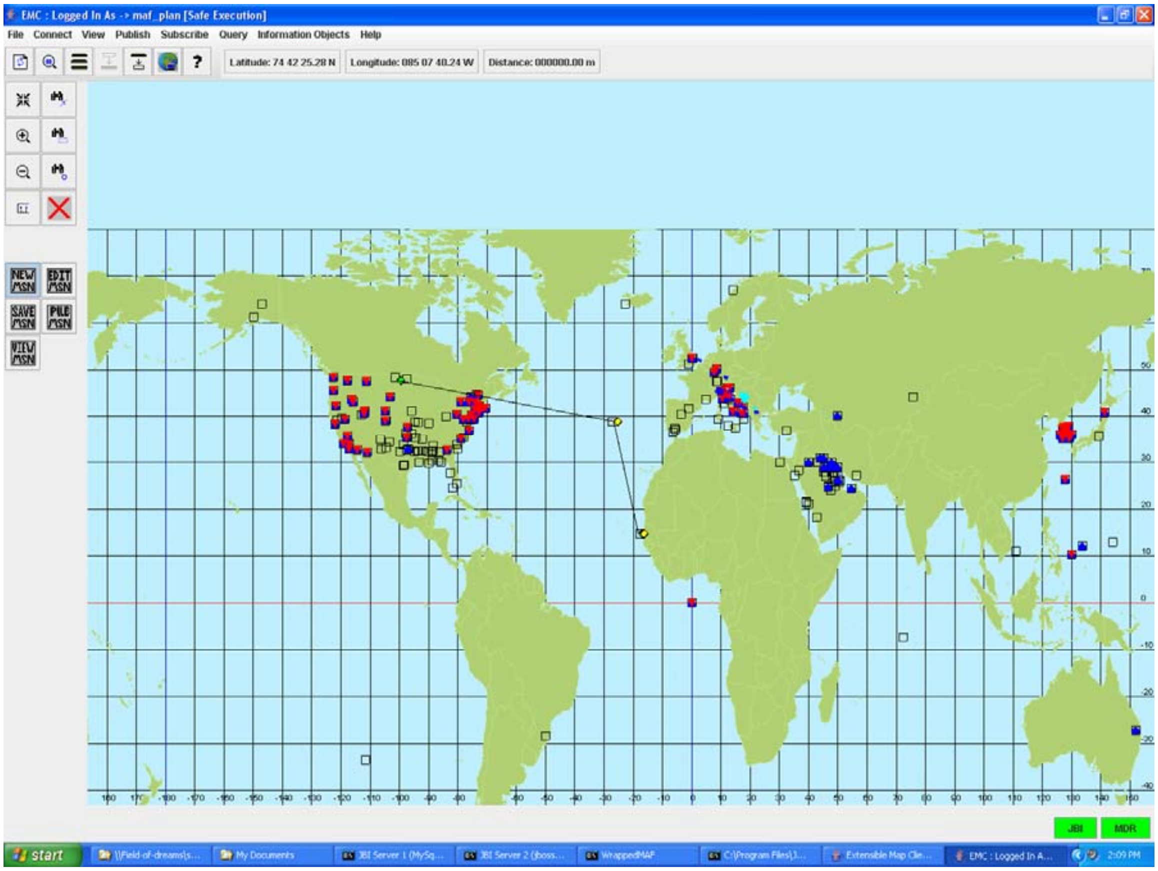


Figure 2: OASIS Dem/Val Scenario

3.4. Misuse Prevention System Infrastructure

The diagram in Figure 4 illustrates the Misuse Prevention Architecture instantiated to the OASIS Dem/Val system. Because the JBI was written in Java, we needed to use a different wrapper technology than we were accustomed to, namely a tool we developed for this project called JavaWrap that mimics our Windows-based wrapping technology. However, that Windows-based wrapper technology, called Safe Family, was also employed to detect operating system-based attacks.


EMC : Logged In As -> maf_plan [Safe Execution]

Latitude: 74 42 25.28 N

Longitude: 085 07 40.24 W

Distance: 000000.00 m

Figure 3: The MAF/CAF GUI

Three different tools were employed in modeling the application and detecting potential harm from these models. The models specify respectively, the semantics of the air tasking order domain, consistency of a specific program data structure, consistency of the program's Mission Plan data Structures, and the domain of legal operating system calls. They will each be discussed individually in the Technical Results sections below.

Most of the agents in the Dem/Val scenario were programmatic “stub code” that published pre-canned information from files, rather than, for example, publishing actual weather data. However, a substantial operator interface illustrated in Figure 3 was provided for the MAF and CAF agents (the same component, actually, just applied to different information at different times for different purposes). Notice the locations of various airfields around the world and the route being constructed from the US to Africa.

These are easily changed by actions of the user before publishing the ATO, using the bottom right button at the left of the screen in the cluster of 5.

Because this interface could be used maliciously by an insider, it was chosen as the component from which our Misuse Detection and Prevention Architecture would protect the system.

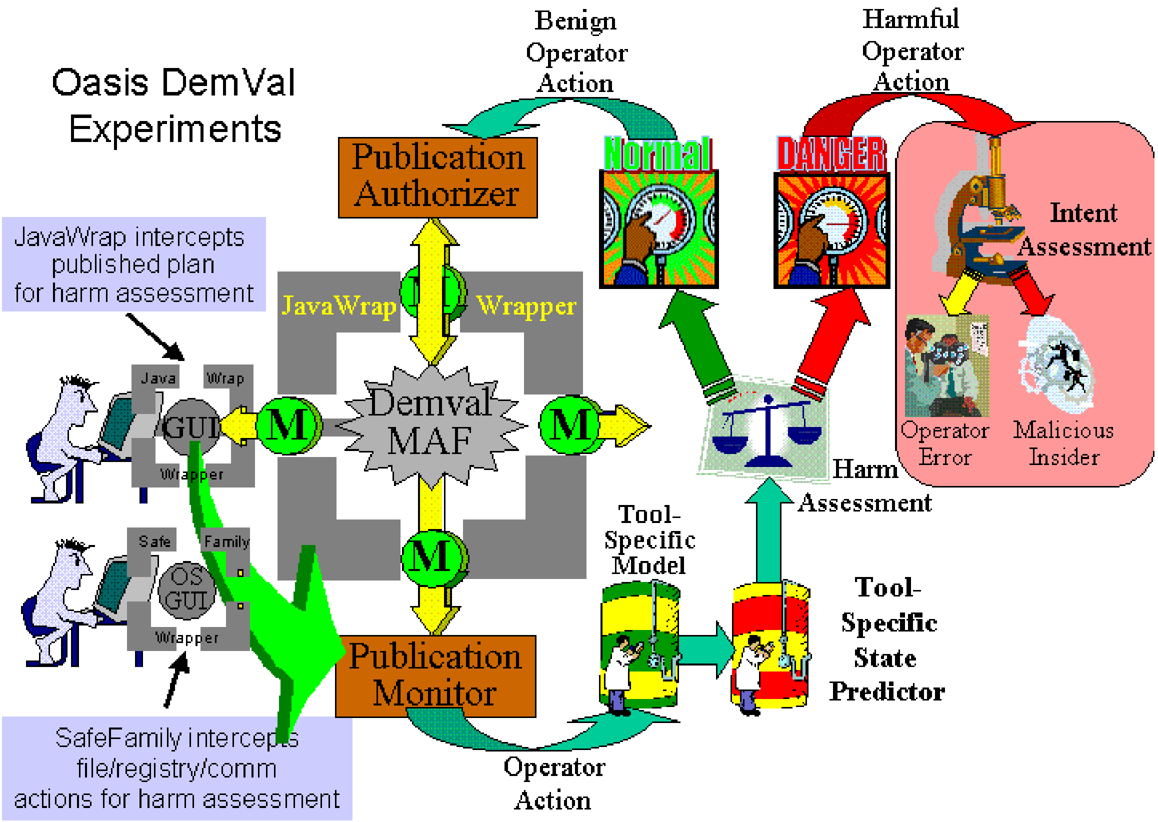


Figure 4: Instrumented Dem/Val System Architecture

In order to produce valid scenarios for the MAF and CAF operators, it was necessary to drive the system to a valid state so that they could use information that was indeed realistic. This was done by specializing the JavaWrapped publications to keep an XML-based history of the publications during a test run of the scenario above. We then built driver code that ran this scenario, publishing it through the normal JBI interfaces, again using specialized wrappers to read the information that we had saved in the history.

4. Technical Result: Impending Harm Detection

Notice in the diagram above that the predicted state of the system can be used to assess impending harm. This state is derived from information from the Java Wrapped version of the JBI that intercepts calls to publish and read information from the repository. The attempts to publish are filtered by our Harm Detector and Malicious Behavior Detector; if they determine that harm will ensue, the publication is blocked.

The operational model we developed for the OASIS system was used to detect corrupted data, effectively data whose use in the final air tasking order would have harmed the mission. The model itself is expressed as a set of rules that constitute the "application semantics."

4.1. *Application Semantics: the* Operational System Model

The following list of rules characterizes the valid air tasking orders that can be formed by the MAF / CAF operators.

1. Planes have types, which have a maximum Range before the plane must land or be refueled (refueling resets the starting point to the refueling point - i.e. assumes the plane has been fully refueled).

2. Planes have types which have a minimum required runway length for takeoffs and landings

3. Planes cannot land or takeoff in restricted-access zones (defined as rectangles aligned with the lat/long axis).

4. Planes have types which can not go to certain destinations

5. Each airport has a minimum turn around time and a plane landing at that airport must not takeoff before that minimum turnaround time has expired

6. Each mission has an objective for that mission's plane and that plane must reach the destination specified in that objective by the time specified in that objective. This objective is associated with the type of the plane.

7. Refueling (defined by the MAF to occur at a point) can only occur in rectangular refueling-areas (aligned with the lat/long axis).

8. Each leg in a mission must get the plane closer to its destination. Offload events (which have end points equal to their start points) do not count as a leg for this rule.

9. A plane's weight (determined by its plane type) cannot exceed the weight-handling maximum for each runway it lands on or takes off from.

10. A plane can only land or take off from a runway at night (1800 to 0600 local time) if that runway is equipped with night lighting.

11. The duration of a leg must exceed the time needed to fly that leg (i.e. the distance between its start and end locations) at the plane's maximum speed

12. Offloads must occur at the same place as the landing that preceded them.

13. Offloads must have a minimum duration based on the type of airplane

14. All missions must start with a takeoff and end with a landing or offload (i.e. no suicide missions).

15. All takeoffs (other than the initial takeoff) must be immediately preceded by a landing or offload.

16. All landings must be immediately preceded by a takeoff, waypoint, or refueling.

17. Each refueling must be immediately preceded by a takeoff, waypoint, or refueling.

18. All waypoints must be immediately preceded by a takeoff, waypoint, or refueling.

19. A landing must immediately precede all offloads.

20. Each leg must start after the end of the immediately preceding leg ends.

21. Each leg must end after the start of that leg.

22. Each takeoff (other than the initial takeoff) must be from the same place as the previous landing.

The rules monitor an operator's behavior within a Dem/Val component (the MAF\CAF) to detect harm at the point that the operator's actions are committed. In Dem/Val this is when the Mission Plan constructed by the operator is published (made visible to other portions of the system). Harm is detected by determining whether the plan satisfies a set of application-specific integrity constraints (such as that a plane must take off from the same airfield that it landed at). If a plan fails any of these integrity checks (indicating that the plan cannot be safely executed), its publication is blocked to prevent that harm. An analysis is then performed on the offending action, the failed integrity check(s), and the history of operator actions that led to the offending action to determine whether there is a consistent pattern of malicious operator activity.

Harmful plans are characterized using a relatively simple rule-based inferencing system and are then archived in a "case-file" that stores and compares several bad plans produced by the same operator. Post hoc reasoning then attempts to determine whether there is a consistent pattern of malicious activity. It is important to note that the reasoning done here is whether the plan specified by the operator is harmful (as opposed to reasoning about whether the system code was compromised in such a way to change the plan requested by the operator). Thus the question is completely whether the operator misused his privilege to create a harmful plan. Unfortunately, the close coupling between the Misuse Prevention Architecture tools required to make such an assessment was never implemented, so assigning blame in the data corruption area was never achieved.

4.2. Rule Violations

The following are sample error messages produced when the rules are violated during actual executions of the CAF.

```
************************************************
>>waypoint-refuel-or-landing-does-not-bring-aircraft-closer-to-destination
 - leg from: 1 to: 2 aircraft: RF5A<<
>>the distance between 3 (LDG) and its final landing 3 is 0.0 miles<<
>>the distance between 2 (REFUEL) and its final landing 3 is 2420.0 miles<<
>>the distance between 1 (TO) and its final landing 3 is 1965.0 miles<<
>>offload-and-refuel-must-have-end-time-after-the-start-time 2<<
>>aircraft-cannot-refuel-outside-refuel-airspace 2
at lat: 43.59375 long: 4.8721513748168945<<
```

4.3. Implementation Details

The rules are matched to an XML representation of the Air Tasking Orders (ATOs) using a parser that parses into the rule language for a tool called Jess. This mechanism is invoked on each parsed XML file and if a constraint represented by a rule is violated, the publication is rejected (the Java Wrapper of the JBI interface code receives an illegal operation exception). Appendix I contains the rules used to describe the rules expressed informally above. Notice that there is a considerable amount of translation and coding required to express these rules in Jess. It should be emphasized that the mechanisms for implementing the various application domain models in this project should be considered to be idiosyncratic to the application and not an intrinsic part of the Misuse Prevention Architecture. Appendix II completes the model by providing the various initialization data elements that are needed to characterize the actual situation on the ground and in the air, the Mission Objectives, Airspaces, Airbases, and Aircraft. These environmental parameters describe capabilities of aircraft, available services of airbases, etc. Together with the actual plan, a temporary database is constructed from these parameters and the rules are matched against it. The rule base is allowed to rely on user-defined functions (in Java) that characterize application-specific predicates that are difficult to compute in a table-driven or inference-driven fashion, such as: illformed-mission-object, illformed-mission-event-row, date-time-in-minutes, and lat-long-distance calculator.

5. Technical Result: Successful Detection of Malicious Operator Attacks

The project focused initially on the above vector of harmful operator behavior in the DemVal system – the publication of malformed Mission Plans which either could not be performed (because they violate execution constraints in the physical world) or which would cause harm if they were performed (by delivering supplies to the wrong location or by interfering with other operations in the same airspace).

A second major accomplishment was to monitor an operator's behavior within a DemVal component (the MAF\CAF) to determine whether the detected harmful behavior was malicious or just an operator error. In DemVal this detection occurs when the Mission Plan constructed by the operator is published (made visible to other portions of the system). Harm is detected by determining whether the plan satisfies a set of application specific integrity constraints (such as that a plane must takeoff from the same airfield that it landed at). If a plan fails any of these integrity checks (indicating that the plan cannot be safely executed), its publication is blocked to prevent that harm. An analysis is then performed on the offending action, the failed integrity check(s), and the history of operator actions that led to the offending action to determine whether there is a consistent pattern of malicious operator activity.

5.1. Inferring Intent

Inferring the actual intent of an operator is a very difficult task. In many cases, an operator could perform a harmful action either by accident or intentionally. Many programs perform a certain amount of validity checking and warn the user that what they are doing seems to be out of bounds; unfortunately, the DEMVAL programs that we examined aren't so equipped. Furthermore, the operator interface is reasonable "clunky"; therefore, a user might easily mistakenly enter invalid and harmful information by accident. Under such circumstances, we decided that the best course of action that we could follow is to log all user actions, to call attention to those that do eventually cause harm, and for those that seem malicious, to open a "case book" on the user, documenting the suspicious behavior and leaving final determinations of intent to human examiners. In the real world, such determinations are complex security and personnel issues, whose final resolution is unlikely to be left solely to computer systems.

However, some behavior is more suspicious than others and so part of our goal is to identify those actions that are more likely to have been the result of malicious intent. We have developed guiding principles for this assessment: A set of actions that is consistent with a plan for causing harm but not consistent with normal operations is cause for suspicion. The larger the deviation between the two (e.g. the number of steps consistent with harmful outcome, but inconsistent with benign outcome) is a metric of how suspicious the activity is.

5.2. Vulnerability Analysis

Attack modeling is the process of systematically enumerating all of the ways in which a computational environment can be attacked and discovering how those attacks can lead to resource compromises. The output of attack modeling is a set of complex, multi-step plans that an attacker might use. The attack plans are hierarchical; each sub-plan corresponds to an attack affecting a compromise that enables other sub-plans downstream.

Vulnerability analysis is a backward chaining goal-directed reasoning process. It begins with desirable properties, finds ways to compromise those resources that deliver these properties and then find ways to enable these compromises. The attack plans developed may be quite complex, multi-stage plans in which one step enables a compromise (e.g. gaining access to a user account) that serves as a foothold for succeeding steps (e.g. monitoring network traffic to steal information).

As part of the PMOP project, we extended the vulnerability analysis core that we have developed previously in the Self-Regenerative Systems (SRS) program to cover cases that arise in the DEMVAL system but that were not within the scope of the earlier system. One example is including the use of normal "SaveAs" dialogs to overwrite system resources.

5.3. Expected Behavior Analysis

Whereas vulnerability analysis is the systematic exploration of ways to compromise a system, expected behavior analysis is the systematic exploration of ways to beneficially utilize the capabilities of a system.

It proceeds as described above in Vulnerability Analysis except that the exploration is through the steps needed to achieve desirable states of the system – such as validating the availability of a resource before attempt to use it.

It should be noted that little work has occurred to date on developing expected-behavior plans because there was no machinery to consume and utilize those plans. However, now with the advent of our *Behavior Monitor,* the user's progress through these plans can be tracked and expectations established about the user's next steps.

These expected-behavior plans can then be used as the "white-hat" dual to the more extensively studied "black-hat" attack plans. Moreover, they should prove highly useful to the cognitive agents programs underway in IPTO as a source of modeling user intent.

5.4. Plan Generation

Expected-Behavior and Attack plans are generated by a rule-based inference system that uses the operational system model to reason about how one might affect a desirable system property – attack plans seek to negatively affect that desirable system property while expected-behavior plans seek to positively affect that desirable system property. Fundamentally, this rule base deals with how different components depend on and control one another. We make this rule base as abstract and general as possible. This puts the notion of control and dependency at the center of the reasoning process.

Both expected-behavior and attack plans are generated in the same formalism, which facilitates the use of a common plan recognizer to recognize instances of these plans to differentiate malicious intent from operator error.

5.5. Plan Recognition

These expected behavior and attack plans are then interpreted by the plan recognition component of the system that is informed by inputs from the full gamut of sensors available – most notably the *Behavior Monitor* that provides the pending application-level user command. It collates these inputs looking for specific patterns of activity characteristic of each step of a plan; for example, high rates of password scanning alarms from an intrusion detection system are characteristic of an early stage of an attack in which the attacker is attempting to gain first access. The plan recognizer maintains a set of active hypotheses; each hypothesis corresponds to a particular expected-behavior or attack plan some of whose steps have already been observed.

5.6. Systematic Monitoring

Pervasive placement of sensors throughout the application environment allows us to collect evidence that is crucial to discriminating between intentional and accidental actions that cause harm. The following are significant capabilities that such monitoring provides:

Document Accountability: Because all modifications to application documents are monitored and attributed to a user, the insider cannot repudiate actions he has taken after the fact, including creation, modification, and deletion of documents, and sending and receipt of messages. Current COTS office products do not have secure audit capabilities, which are essential for investigating and mitigating insider threats.

5.7. An Example of Intent Analysis

In this project, we focused primarily on the MAF editor component of the DEMVAL system. This is an interactive graphical editor for mission plans. Interactive editing systems present a very difficult context for intent analysis, because users often make mistakes and some of these are left uncorrected. This is particularly true for the MAF system, which it turns out is a prototype system that performs very little validity checking of its own and that therefore provides little coaching of the user. However, there is a specific case that fits our overall notion of suspicious behavior and that is an editing action to an already valid plan to one that is harmful. Such actions can easily occur within the DEMVAL system, because mobility plans (which is what the MAF editor operates on) are first created by an operator in CONUS and then edited by an operator in theater. This second operator has the opportunity to change a correct plan to one that results in harm.

This analysis process is invoked whenever the higher levels of the system detect a harmful action. The analysis begins by consulting a trace of operator actions; these are captured in an XML formatted log of method invocations, like that shown below:

```
   missing-leg 5 6
   **end-of-messages**
   <trace>
   <MethodEnter
   method
Class="mil.af.rl.jbi.client.ExtensibleMappingClient.toolsets.MissionPlan
.MissionEventObject"
      thread="0"/>
   <MethodReturn
```

```
      method
Class="mil.af.rl.jbi.client.ExtensibleMappingClient.toolsets.MissionPlan
.MissionEventObject"
      thread="0">
     <this
class="mil.af.rl.jbi.client.ExtensibleMappingClient.toolsets.MissionPlan
.MissionEventObject"
      printer="1"/></MethodReturn>
   <MethodEnter methodName="setInformation"

methodClass="mil.af.rl.jbi.client.ExtensibleMappingClient.toolsets.Missi
onPlan.MissionEventObject"
    methodSignature="(Ljava/lang/String;Ljava/lang/String;)V" thread="0"
    arg0="EVTTYPE" arg1="TO">
     <this

class="mil.af.rl.jbi.client.ExtensibleMappingClient.toolsets.MissionPlan
.MissionEventObject"
        printer="1"/></MethodEnter>
```

The XML log is parsed and then interpreted, method by method. During this interpretation we build a model of the effect of the method calls on the MAF system's internal mission data structures as shown below:

```
(("missing-leg 5 6"
 (EVENT :THIS ("MissionEventObject" "1") :EVTTYPE "TO" :EVTCD "I" :EVTSEQID "1" :LOCID "KBLV-1"
  :LATITUDE "-89.804" :LONGITUDE "38.671" :TIMEON "2004-05-27T19:25:23Z" :TIMEOFF
  "2004-05-27T19:25:23Z" :ALT "0" :AMCPURPCD "A" :EVTSUBTYPE "-" :SUBTYPECALLSIGN "-" :SUBTYPEFREQ "-
"
  :SUBTYPEMSNCD "-")
 (EVENT :THIS ("MissionEventObject" "2") :EVTTYPE "REFUEL" :EVTCD "T" :EVTSEQID "2" :LOCID "PATRIOT-2"
  :LATITUDE "3.164" :LONGITUDE "52.031" :TIMEON "2004-05-28T03:05:20Z" :TIMEOFF "2004-05-28T03:05:20Z"
  :ALT "280" :AMCPURPCD "Z" :EVTSUBTYPE "-" :SUBTYPECALLSIGN "-" :SUBTYPEFREQ "-" :SUBTYPEMSNCD "-
")
 (EVENT :THIS ("MissionEventObject" "3") :EVTTYPE "LDG" :EVTCD "I" :EVTSEQID "3" :LOCID "LIPA-3"
  :LATITUDE "12.070" :LONGITUDE "46.230" :TIMEON "2004-05-28T04:45:20Z" :TIMEOFF
  "2004-05-28T04:45:20Z" :ALT "0" :AMCPURPCD "A" :EVTSUBTYPE "-" :SUBTYPECALLSIGN "-" :SUBTYPEFREQ "-
"
  :SUBTYPEMSNCD "-")
```

Special notice is taken during this process to identify the first action that causes the plan to become harmful in the way flagged by the higher levels of the system. Actions that explicitly edit out aspects of the plan that had previously been valid are flagged for special consideration, as is illustrated below:

```
MISSING-LEG Between event 5 and 6
CREATING event 1 Take Off       05/27/2004 19:25:23   KBLV   -89.80  38.67
CREATING event 2 Refuel   05/28/2004 03:05:20 PATRIOT 3.16   52.03
CREATING event 3 LDG      05/28/2004 04:45:20    LIPA 12.07  46.23
CREATING event 4 Take Off       05/28/2004 07:20:20   LIPA   12.07   46.23
CREATING event 5 LDG      05/28/2004 08:35:20    LICZ 14.73  37.62
CREATING event 6 Take Off       05/28/2004 11:35:20   LICZ   14.73   37.44
CREATING event 7 LDG      05/28/2004 17:15:20    OEKH 47.70  24.08
EDITING  event 6 Take Off       05/28/2004 11:35:20   LICZ   5.43    47.64
Editing event after its creation
```

Not leaving from where you landed 5 6 14.726 37.617 5.4346514 47.63672

Editing over existing leg causes error - Malicious

6. Technical Result: Resource-Corruption Attacks Defended

In addition to the Corrupted Mission Data and Malicious Operator Detection mechanisms for detecting and preventing insider attacks through install-base software – for which application models relating user actions to application operations exist – we provided a Resource-Corruption Detection mechanism to detect and prevent insider attacks through other installed software.

The chief reason for both legitimate users and malicious insiders to write their own programs is to eliminate the manual labor and time required to interactively invoke a structured sequence of operations. Malicious insiders may also script an attack in the hope of inflicting more damage before the attack is discovered or to activate it at a time when they are offsite and cannot be easily apprehended.

No matter why it was written, an executing program can only attack a system in two ways: it can either use that system's interfaces and invoke operations available through those interfaces, or it can attack the resources required by that system. Protection of attacks of the former type has been described above.

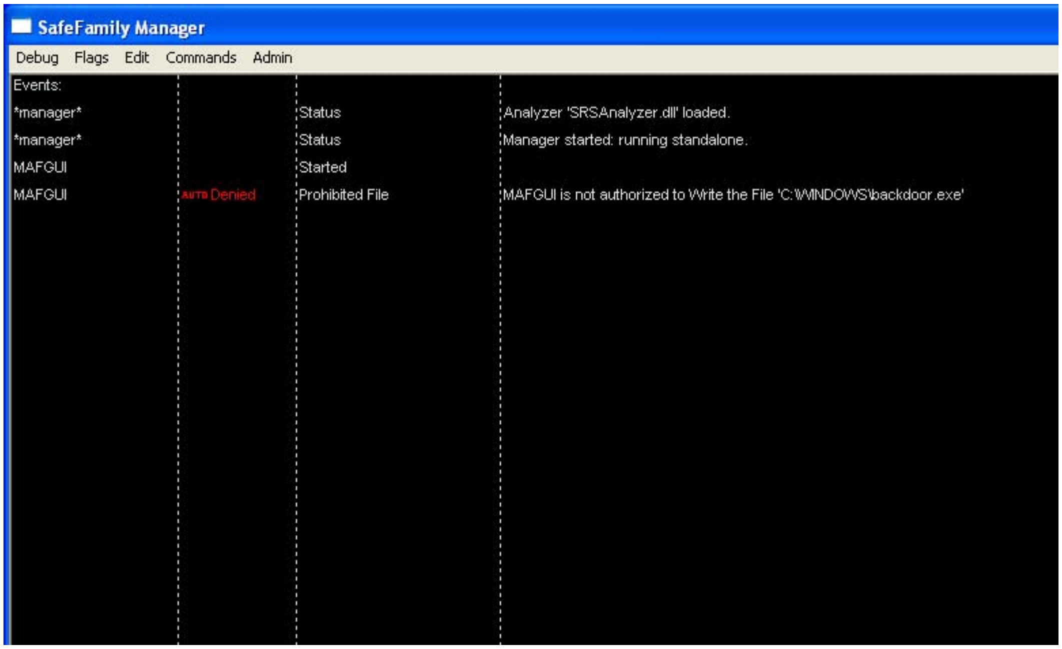
In the latter case, the application is not attacked directly, but instead indirectly through the resources on which it depends. These resource attacks prevent the application from using a required resource by destroying it (e.g. deleting a file or registry-key), corrupting it (e.g. replacing a file with an old or malformed version), or occupying it (e.g. allocating it to some other non-critical task).

To detect and block this class of insider attack, the set of resources on which the application depends must be identified, and the APIs that control those resources must be wrapped and mediated (as we had already done in our Safe Family wrappers [Balzer, Balzer and Goldman]). Resource control operations invoked by the executing program can then be fed into and evaluated by the Misuse Prevention Architecture with harm assigned to those operations that remove or reduce the accessibility of resources on which the application depends. Role-based resource-control plans are used to identify nominal behavior and distinguish it from malicious attacks.

It should be noted that some of the system level resources (such as files and registry-keys) can be accessed through software commonly available in a site's install base (such as the Windows Explorer and the Registry Editor). As such, some of these resource attacks could be mounted through that install base software without installing any additional software. We actually delayed consideration of all resource attacks until we could incorporate them into this package, because their detection and blockage relies upon the same set of resource-level wrappers discussed above, whether they are mounted from install base or newly installed software.

In order to protect from this malicious behavior we integrated our Safe Family wrappers (developed on prior DARPA projects) into the Misuse Prevention Architecture as an additional behavior monitor that monitors operator behavior and prevents harm to the application's or system's resources. In the MAF application this could occur only through the use of its SaveAs GUI, but we also used this mechanism to protect against the operator's malicious use of the operating system's GUI (the Explorer process) to harm the application's or system's resources.

Impending Harm was defined through the Safe Family's rule language as any deletion or modification of an application or system resource. While the former is well-defined (for Dem/Val all application resources are contained within a single directory sub-tree) the latter is very ill-defined with widely scattered resources that could affect system and/or application operation. We therefore conservatively defined harm as deleting or modifying any existing file (i.e. the SaveAs GUI can only be used benignly to create new files). It should be noted that this conservative definition of harm is not at all influenced by the Misuse Prevention Architecture, but rather results entirely from the difficulty in identifying the scope of operating system resources that need to be protected.



```
Events:
*manager*                      Status           Analyzer 'SRSAnalyzer.dll' loaded.
*manager*                      Status           Manager started: running standalone.
MAFGUI                         Started
MAFGUI         AUTO Denied     Prohibited File  MAFGUI is not authorized to Write the File 'C:\WINDOWS\backdoor.exe'
```

Figure 5: Thwarted Attempt to Write File

Figure 5 illustrates the prevention of such a misuse in the MAF. Notice that the MAFGUI is being prevented (AUTODenied in red) from writing the file C:\WINDOWS\backdoor.exe.

In addition to blocking via the operator's use of the "SaveAs" GUI, we added a final vector for malicious operator behavior detection in our chosen target application (Dem/Val) – the use of the operating system GUI and the programs it can launch to cause harm to the chosen target application.

To protect this operating system vector, where the malicious insider is operating outside the protected application with the objective of corrupting or disabling the protected application, we again used the Safe Family wrappers to wrap the GUI itself (actually the Windows Explorer process) so that all behavior produced by that process, and all processes spawned from it, could be monitored to block harm to the protected application.

7. Measurements of Success

7.1. Satisfying Self-Regenerative Systems (SRS) Program Objectives

Although the insider threat is defined to operate inside the security perimeter of the system, we chose to deal with the insider threat within the context of a "cognitively immune" system that is prepared to deal with threats coming solely from the inside, as well as threats from coordinated actions of insiders and outsiders. This is because our model-based harm assessment naturally handles the case of unauthorized users being granted access to a protected resource as harmful.

This “cognitively immune” system satisfies not only the specific objectives of the Insider Threat area, but also many of the overall SRS program objectives:

7.1.1. Thwart or Delay at least 10% of Insider Attacks

Because our attack detectors are based on model-based prediction of harm to the system, no matter how stealthy and convoluted the insider attack, it has to eventually harm the system (otherwise it would not be an attack) and when it does, that step will be detected before being acted upon by the system because of the predicted harm to the system. This assumes a high fidelity model (so that the predicted effects of an action are accurate) and attacks that cause harm relatively immediately (more precisely within the look-ahead horizon of our reasoner) – i.e. putting the system into a state in which harm was inevitable or could be forced by an adversary beyond the reasoner’s look-ahead horizon.

Because neither of these assumptions can be fully satisfied (our operational models could never exactly model the protected system and our reasoner necessarily has a limited look-ahead), we can never detect 100% of the insider attacks. Nevertheless, with the modeling and reasoning capabilities we produced, the vast majority of insider attacks were detected and blocked – far exceeding the requested 10% minimum. (See the Red Team Experiment section below.)

7.1.2. Learn its own Vulnerabilities over time

Our system maintains a trust model indicating the degree to which its computational resources are believed to be compromised. We extend this trust model to also include models indicating the degree to which the system has reason to doubt the "bone fides" of trusted insiders. In both cases the trust model represents information learned from previous experiences.

The system is reflective, i.e. it has models of its own behavior from which it can make predictions about the outcome of an action it is considering taking. To deal with insider threats we extend the modeling framework to include models of the other entities that will be affected by the system's action. For example, if the software system is a logistics management system, we provided (admittedly fairly simple) models of how the motion of fuel (for example) will impact the air tasking plan.

The system detects deviations from intended system behavior. In the case of the insider threat, we extended the notion of deviation from intended behavior to include behavior that is allowable, but nevertheless harmful in the current context.

When deviations from intended behavior are detected the system diagnoses the cause and updates its trust model. In the case of an insider threat, this means that when actions are taken that lead to bad effects, the system must determine the full scope of actions that led to the bad effect and must then attribute (partial) blame to each of the participating (computational or human) entities.

7.1.3. Ameliorate those Vulnerabilities

The system has models of its own vulnerabilities and of attacks that can exploit these. In the case of the insider threat we look at the attack plans in a new light. In general, attack plans are complex sequences of actions that acquire greater privileges for an outsider, finally allowing him to compromise a critical resource. An insider, working in concert with an outside, can short-circuit such a plan, because the insider can grant the access to the outsider that would have otherwise required exploitation of vulnerabilities (e.g. buffer overflows, used to gain privilege, can be replaced by the insider granting the privilege). We could modify our vulnerability analysis tools to include the use of trusted insiders in addition to traditional exploitations of vulnerabilities. The resulting attack plans would then represent action sequences taken by combinations of insiders and outsiders that ultimately compromise a mission critical resource (e.g. deleting critical data from or adding incorrect data to a protected database). The same plan recognition techniques that track the evolution of an outsider attack could then track these combined insider-outsider attacks and update the trust model as increasing evidence of ill intent is inferred.

7.1.4. Regenerate Service After Attack

The system reconfigures and regenerates itself as it observes signs of compromise. One role of the system's models and the instrumentation of the executing code is to dynamically provision redundant information from which it can regenerate an uncompromised, consistent and reasonably current image of itself. As signs of compromise are noticed, these backup copies of information are substituted for current copies if the overall estimate is that the system will better deliver services to its critical users if the regeneration is undertaken. Other SRS projects will provide sophisticated techniques for the gathering and protection of such data; our contribution is in recognizing when it is useful to roll back. In the case of insider threats, this may be extended to include the gradual (or sudden) revocation of privileges enjoyed by a previously trusted insider as evidence mounts that the trust is not warranted. In more extreme case, this may take the form of "fish bowling" the suspect user, allowing him to operate against shadow data that has no effect on the real world, while feeding back data indicating the opposite. All the machinery necessary to do this is implicit in this framework, although it was more ambitious than we could attempt during this initial effort.

In the Dem/Val context, there is really no issue of regeneration, because we block harm before it takes effect. We could consider black-listing or grey-listing the operator based on suspicion as a natural extension of this effort. However, in this context there was little need to provide a recovery action.

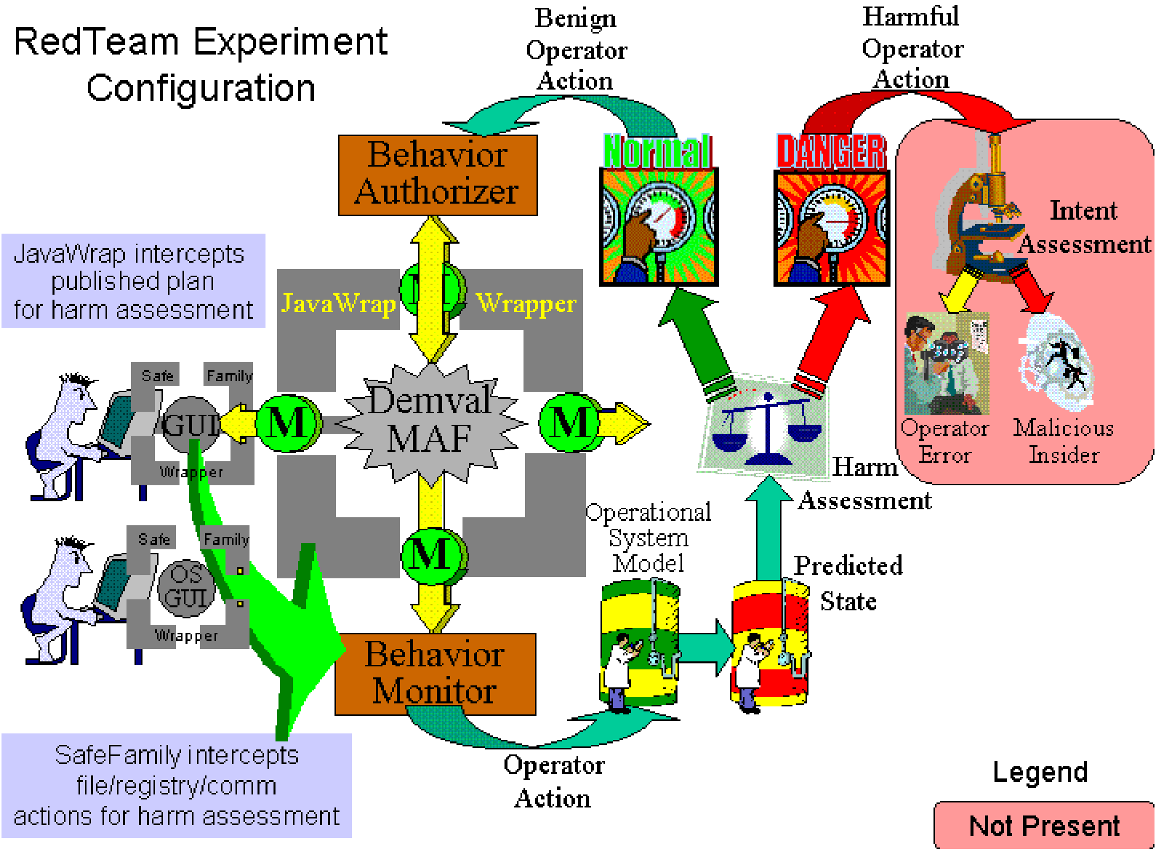


Figure 6: Red Team Configuration

7.2. Validation: Red Team Experiment

7.2.1. An Experiment to Determine Ability to Thwart Insider Attack

In order to validate our system's ability to detect and thwart insider attacks, an experiment was conducted that allowed malicious attacks to be designed to achieve any of the following three "flags":

1. Attackers ability to harm application using only the CAF application GUI (SaveAs/Open GUI excluded). We used the jointly-defined subset of application semantics described under the Corrupted Mission section above (Technical Result 1).
2. Ability to harm the application using only the SaveAs/Open GUI
3. Ability to harm the application using the OS GUI (the Explorer process), while running other programs was disallowed.

The system was configured as indicated in Figure 6. Notice that the assessment of malicious vs. accidental intent was not present in this version of the system.

7.2.2. Red Team Experiment Results

Out of 14 attempts to harm the application or induce a false positive, the following successful attempts were registered. In addition the number of false positives where the system erroneously detected harmful behavior are included.

1. Harm application using only application GUI [0 harm, 1 false positive]
2. Harm application using only SaveAs/Open GUI [0 harm, 0 false positive]
3. Harm application using OS GUI (Explorer process) [1 harm, 0 false positive]

Inasmuch as the goal of the SRS program was to achieve a 10% success rate, this was clearly well beyond the expected goal, no matter how one integrates the combinations of attacks and false positives.

7.3. Future Improvements

It is notable that a careful choice of flags to cover the space and focus the attacks was partially responsible for our success. To improve our results in the future we would need to contend with the following.

- We would need to protect a more realistic target system. The choice of this system was flawed from the aspect of our being unable to find experts to provide suitable semantics for the MAF client. In addition, the semantics used here protect the future of a planned mission, not an aspect of the system itself that will fail.

- Integrate with program misbehavior sensors. The system as it resulted was fragmented into pieces that accomplished the various technical results above. It was never integrated into a uniform whole in which the pieces of knowledge from one were passed to the others. This would be necessary to improve blame assignment, for example.

- Finally, the OASIS system was rather imperfectly implemented, so whenever a program bug arose that corrupted the published plan, the operator was (incorrectly) blamed for the harm, rather than the system defect. Detecting this problem is a significant research issue unto itself.

Finally, several aspects of our approach were never sufficiently tested, largely because of the lack of expertise to provide suitable error-filled operator plans and plans of attack for this system. (This is certain to have stymied the Red Team attackers somewhat as well.) Hence, our mechanisms for matching harmful actions against such plans were only lightly exercised.

8. Conclusions: an Infrastructure for Self Generation

A complete version of the Misuse Prevention Architecture has been developed and is working. This is a plug-in architecture with defined and working plug-ins for the Behavior Monitor module (which monitors operator actions), the State Predictor module (which uses the Operational System model to predict the state which would result from the operator action) the Harm Detector module (which determines whether the predicted state would violate any of the conditions defined in the Harm model), the Behavior Authorizer module (which allows benign operations to be performed and blocks harm causing ones), and the Intent Assessment module (which determines whether harm causing operations were malicious or operator errors).

Together the Misuse Prevention Architecture and these plug-in modules function as a complete end-to-end system for detecting misuse and blocking the harm that would result from such misuse.

MAF/CAF is a plan editing system, so the particulars we focused on were tailored to its domain of air plan construction, but the approach and methodology described here is far more generic. It is an application of the general methodology of vulnerability analysis. The first question we begin with is:

8.1. How can a plan be harmful?

- It can fail to achieve its mission goal

- It can achieve other goals than the mission goal which are harmful.

- A flaw in it can impede other parts of the planning process.

- It can only be harmful at the point that it is put into action.

In MAF/CAF the natural notion of a plan being put into action is when it is published. A plan for running a continuous process might be treated differently, namely by constant monitoring during its execution.

We deal with each of these categories of harm in the order listed above: There are three main categories of failing to achieve the mission goal:

1. The plan can be structurally inconsistent. In the Air Mobility domain, structural inconsistency means that successive legs of a plan are violations of basic physical rules as applied to the plan: Each leg must begin from where the last one left off, each leg must start after the last one ended and each leg must end after it starts. These constraints can be expressed as a set of data structure consistency rules that are checked after every operator action.

2. The plan can be unexecutable: This involves checking at a more semantic level and making sure that the actions of each step are within the capability of the resources tasked to that action: This includes that planes have the range to go from source to destination (if there is refueling then that constitutes the end of a leg), they have the capacity to carry the cargo assigned to them, they have the speed to traverse the leg in planned time, etc. Checking these rules requires detailed data of the specific vehicles. For this project, we instead just made up the appropriate data tables.

3. Even if the syntactic criteria of 1 and the semantic criteria of 2 are met, the plan may fail to achieve the goal simply because the final step is not the intended goal. We did not actually have this information either, but we were able to make it up for the Red Team experiments (see Appendix II).

8.2. *What Are the Consequences of Harm?*

The second category of harm is harmful side-effects of a plan. There are two major sub-categories:

- Harm to the crew
- Harm to friendly and neutral elements.

Checking these involves modeling the environment in which the plan is executed and reasoning about consequences such as:

- Flying a plane through a danger zone (e.g. downwind of a WMD plant that is targeted)
- Taking an action that causes direct harm to friendly or neutrals: Bombing one of them
- Taking an action that causes indirect harm to friendly or neutrals: Bombing a WMD plant that is upwind of a population area.

The CAF editor actually subscribes to chemical-hazard and weather analysis data so it could use these to do the reasoning suggested above. However, for our own purposes it was efficacious to make up the data.

The Misuse Prevention Architecture is based on the idea of preventing harm created (at least partially) by an insider who already possesses the privileges needed to cause harm. The insider is in many ways very much like the outsider, except that the outsider must take prior steps to acquire the privilege necessary to cause harm. There is an additional role the insider may play: as the operator of a system to which there is no external access.

The third category of harm is that the operator can harm the planning process by intentionally filing an invalid plan. We will block this from being published, but the failure to complete this step of planning process can itself be the attack, if it prevents the rest of the planning process from finishing. However, the operator could achieve the same effect by simply not publishing a plan at all. Our intervention – preventing an invalid plan from being published – provides an earlier alert for the rest of the system and avoids its having to discover this problem later when less time exists to react to the problem.

8.3. Broadening the Vulnerability Analysis

So far we have concentrated on the actions of the insider as the operator of a system that is inaccessible to an outsider because it is a unique vulnerability. However, the operator is typically at least a normal user of the host systems as well. Thus we can broaden the vulnerability analysis to ask:

How can the operator cause a harmful plan to be generated using his privilege as a system user? Vulnerability analysis leads to a variety of answers such as:

- The operator can modify a file holding the representation of a plan being worked on by some other operator before it's published by that operator.
- The operator can modify a file containing input data of the system.
- The operator can modify a CLASS file so that it high jacks the program and modifies an otherwise harmless plan to be harmful.
- The operator can modify a source file so that once it is compiled in a normal system, rebuilding it will high jack the program as in the previous step.

Notice that the last two of these are particularly insidious since the program high jacked might be that of another user, making it appear that the other user caused the problem.

To deal with this class of problems one can:

- Impose a set of "System Wrappers" that detect operator actions to the file system etc. These log the operators action but do not block them in those cases (which are nearly all the cases) where the operator is doing something within his approved privilege (i.e. the operator may have the privilege to edit source files, so we cannot block that action)

- Monitor the program action to check that no bad plan is published and that each operator request is faithfully carried out by the program.
- Use plan recognition technology to recognize that previously logged operator actions fit within a plan that ultimately causes harm (e.g. cases 3 and 4 above).
- Use the complexity and difficulty of the plan as a measure of operator intent. The more difficult the overall plan, the less likely it is to have been an accident.

8.4. Coordinated Attacks

A final category of insider attack is one that involves coordination with an outsider. For example, the insider changes the access rights to a file, which enables an outsider to change it in a way that leads to a harmful plan being generated. The steps outlined above will deal with such mixed cases as well.

8.5. Conclusion

We strongly feel that the Misuse Prevention Architecture can be adapted to protect other processes in different application domains. It is a plug-in architecture with defined and working plug-ins for the Behavior Monitor module (which monitors operator actions), the State Predictor module (which uses the Operational System model to predict the state which would result from the operator action) the Harm Detector module (which determines whether the predicted state would violate any of the conditions defined in the Harm model), the Behavior Authorizer module (which allows benign operations to be performed and blocks harm causing ones), and the Intent Assessment module (which determines whether harm causing operations were malicious or operator errors).

Together the Misuse Prevention Architecture and these plug-in modules function as a complete end-to-end system for detecting misuse and blocking the harm that would result from such misuse. The primary requirements for reuse entail:

- Modeling the system states and valid transitions;
- Modeling activity sequences of the potential insiders;
- Classifying which sequences are relevant to the insiders’ roles and which represent attacks.

These are highly non-trivial tasks that should only be undertaken for extremely important applications.

9. Key Personnel

In addition to the project's Principal Investigators, Howared Shrobe and Robert M. Balzer, several other researchers on their respective staffs at Massachusetts Institute of Technology and Teknowledge Corp. participated. Robert Ladaga of MIT was instrumental in the Common Lisp application development and algorithm development for the trust models and assessment. David. Wile and Alexander Egyed built the infrastructure for coordinating and visualizing the activities of the various OASIS agents and the analyzers. Neil Goldman built the Java wrapper tool used to intercept the JBI services calls. Tim Hollebeek provided the wrapped Windows system call defenses to intercept malicious file resource attacks. Rand Waltzman wrote the fine-grained analyzer to detect the deceptive flight plans and Marcelo Tallis wrote the JBI driver code and the detailed MAF / CAF GUI wrappers.

10. References

[Anderson, et. al.] Anderson, Debra, Teresa F. Lunt, Harold Javitz, Ann Tamaru, Alfonso Valdes, "Detecting unusual program behavior using the statistical component of the Nextgeneration Intrusion Detection Expert System (NIDES)", Computer Science Laboratory SRI-CSL 95-06 May 1995.

[Axelsson] Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In 6th ACM Conference on computer and communications security, pages 1--7, Kent Ridge Digital Labs, Singapore, 1--4 November 1999.

[Balzer and Goldman] Robert Balzer and Neil Goldman: Mediating Connectors: A Non-ByPassable Process Wrapping Technology, DARPA DISCEX Conference 2000, Hilton Head SC, Jan 25-27, Vol II, pp 361-368.

[Balzer] Robert Balzer: Assuring the Safety of Opening Email Attachments, DARPA DISCEX Conference 2001.

[Barbara] D. Barbara, et al., ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection. SIGMOD Record 2001

[Brooks] Rodney Brooks. The Intelligent Room Project. In Proceedings of the Second International Cognitive Technology Conference (CT'97), Aizu, Japan, August 1997.

[Clemm] Geoffrey Clemm. ODIN -- An Extensible Software Environment. University of Colorado, Boulder, Technical report: CU-CS-262-84. 1984.

[Clemm and Osterweil] Geoffrey Clemm, Leon J. Osterweil: A Mechanism for Environment Integration. TOPLAS 12(1): 1-25 (1990)

[DASADA] DARPA. http://www.rl.af.mil/tech/programs/dasada/

[DoD] Requirements for Ada Programming Support Environments: STONEMAN. United States Department of Defense, Office of the Under Secretary of Defense for Research and Engineering. Feb. 18, 1980. NTIS-AD-A100 404/3.

[Donzeau-Gouge et. al.] Donzeau-Gouge, V., Kahn, G., Lang, B., and Mélèse, B. “Document Structure and Modularity in Mentor,” Proceedings of the ACM SIGSOFT/SIGPLAN Software Symposium on Practical Software Development Environments (1984), pp. 141-148.

[Eckmann, et. al.] Steve T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer, 2000. "STATL: An Attack Language for State-based Intrusion Detection". Dept. of Computer Science, University of California, Santa Barbara.

[Egyed and Balzer] Egyed, A. and Balzer, R.: "Unfriendly COTS Integration - Instrumentaiton and Interfaces for Improved Plugability," Proceedings of the 16th IEEE International Conference on Automated Software Engineering (ASE), San Diego, USA, November 2001.

[Egyed and Wile] A. Egyed and D. Wile. Statechart Simulator for Modeling Architectural Dynamics In Proceedings of the Working IEEE/IFIP Conference on Software Architecture, Amsterdam. Aug 2001. 87-96.

[Forest, et. al.] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for unix processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, pages 120--128, Los Alamitos, CA, 1996.

[Fraser et al] Timothy Fraser, Lee Badger, Mark Feldman: Hardening COTS Software with Generic Software Wrappers. IEEE Symposium on Security and Privacy 1999: 2-16.

[Garlan et al] David Garlan, Robert T. Monroe, and David Wile. ACME: An Architecture Description Interchange Language. Proceedings of CASCON'97 (November, 1997). (See also: http://www.cs.cmu.edu/~acme)

[Garvey and Lunt] T.D. Garvey and T.F. Lunt. Model-based intrusion detection. In Proceedings of the l4th National Computer Security Conference, October 1991.

[Ghosh] Anup Ghosh. “Sandboxing Mobile Code Execution Environments”, URL: http://www.cigital.com/research/sandboxing.html

[Goldman and Balzer] N. Goldman and R. Balzer. The ISI Visual Design Editor Generator. IEEE Symposium on Visual Languages. Tokyo. Sep. 1999. 20-27.

[Hall et. al.] R.S. Hall, D. Heimbigner, and A.L. Wolf. A Cooperative Approach to Support Software Deployment Using the Software Dock. Proc. of ICSE'99: The 1999 Int'l Conf. on Software Engineering, Los Angeles, CA, May 1999, pp. 174-183.

[Harel] D. Harel. Statecharts: A Visual Formalism for Complex Systems Science of Computer Programming, vol. 8, 1987.

[van der Hoek et. al.] A. van der Hoek, D. Heimbigner, and A.L. Wolf. A Generic, Peer-to-Peer Repository for Distributed Configuration Management. Proceedings of the 18th International Conference on Software Engineering, Berlin, Germany, March 1996.

[Hofmeyr] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. Journal of Computer Security, 6:151--180, 1998.

[Holzhauer] Doug Holzhauer and Carrie Kindler. OASIS Scenario Initialization and Execution Guide. Patrick Hurley, ed. AFRL / IF & ITT Industries. April, 2004, 1-38.

[Hunt and Brubacher] Galen Hunt and Doug Brubacher. Detours: Binary Interception of Win32 Functions. Proceedings of the 3rd USENIX Windows NT Symposium. Seattle, WA, July 1999. USENIX.

[Ilgun and Porras] K. Ilgun, R. Kemmerer, and P. Porras. State Transition Analysis: A RuleBased Intrusion Detection System. IEEE Transactions on Software Engineering, 21(3), Mar. 1995.

[Kaiser] Gail E. Kaiser: MARVEL 3: 1: A Multi-User Software Development Environment. International Symposium on Logic Programming, 1993: 36-39

[Khare et.al.] Rohit Khare, Michael Guntersdorfer, Peyman Oreizy, Nenad Medvidovic, Richard N. Taylor. “xADL: Enabling Architecture-Centric Tool Integration With XML.” In Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34), Maui, Hawaii, January 3-6, 2001.

[Kumar and Spafford] Kumar, S. and Spafford, E., "A Pattern Matching Model for Misuse Intrusion Detection," Proceedings of the Seventeenth National Computer Security Conference, pp. 11--21 (Oct. 1994).

[Lee and Stolfo] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, 1998.

[Micallef and Clemm] Josephine Micallef, Geoffrey Clemm: The Asgard System: Activity-Based Configuration Management. International Workshop on Software Configuration Management, 1996: 175-186.

[Oram and Talbott] A. Oram. and S. Talbott. Managing Projects with make. (Feb, 1993).149 pp.

[Porras and Neumann] P.A. Porras and P.G. Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In Proceedings of the Nineteenth National Computer Security Conference, pages 353--365, Baltimore, Maryland, 22-25 October 1997. NIST/NCSC.

[Reiss] Reiss S.P. FIELD: A Friendly Integrated Environment for Learning and Development. Kluwer Academic Publishers, 1994.

[Reps and Teitelbaum] T. W. Reps and T. Teitelbaum. The Synthesizer Generator. Springer-Verlag, New York (1988).

[Ryan et.al] Ryan, J. , Lin,M., and Miikkulainen, R. (1998). "Intrusion Detection with Neural Networks" . In Advances in Neural Information Processing Systems, vol. 10, MIT Press. 1998.

[Tallis and Balzer] Marcelo Tallis and Robert Balzer: Document Integrity through Mediated Interfaces, submitted to DARPA DISCEX Conference 2001.

[Teitelbaum and Masinter] W. Teitelbaum and L. Masinter. The Interlisp Programming Environment, IEEE Computer. April 1981.

[Tichy] W. F. Tichy. RCS: A System for Version Control. Software Practice and Experience. 15(7): July 1985. 637-654.

[Wagner and Soto] D. Wagner and P. Soto. Mimicry attacks on host based intrusion detection systems. In Proc. Ninth ACM Conference on Computer and Communications Security, 2002.

[Wenke and Kaiser] Wenke Lee, Gail E. Kaiser: Interfacing Oz with the PCTE OMS: A Case Study of Integrating a Legacy System with a Standard Object Management System. Journal of Systems Integration 9(4): 329-358 (1999)

[Wile, 1993] D. Wile. Popart: Producers of Parsers and Related Tools, Reference Manual. USC/Information Sciences Institute, Marina del Rey, CA (1993).

[Wile, 2001] D. Wile. Using Dynamic Acme. In Proceedings of a Working Conference on Complex and Dynamic Systems Architecture. Brisbane. Dec. 2001.

[Zeller] Zeller: Versioning System Models through Description Logic. Proc. 8th International Symposium on System Configuration Management (SCM-8), Brussels, July 1998.

Appendix A: Jess Rules for Mission Data Corruption Detection

```
(deftemplate MISSION_EVENT_ROW
       (slot ALTITUDE)
       (slot AMC_PURPOSE_CD)
       (slot EVENT_CD)
       (slot EVENT_SEQ_ID)
       (slot EVENT_SUB_TYPE)
       (slot EVENT_TYPE)
       (slot LOCATION_ID)
       (slot LATITUDE)
       (slot LONGITUDE)
       (slot PLANNED_TIME_OFF)                              ; date time
       (slot PLANNED_TIME_ON)                               ; date time
       (slot SCHEDULED_OFFLOAD)
       (slot SUB_TYPE_CALLSIGN)
       (slot SUB_TYPE_FREQ)
       (slot SUB_TYPE_MSN_CD)
       (slot prev (default -1))
       (slot next (default -1))
       (slot prev-landing (default -1))
       (slot next-landing (default -1))
       (slot distance-since-last-refuel (default -1))
       (slot inside-refueling-area (default "NO"))
       (slot uuid)                                                 ; make sure
there cannot be duplicates
)

(deftemplate MISSIONOBJECT
       (slot ABPID)
       (slot MSNNO)
       (slot AMCMSNNO)
       (slot TSKUNITID)
       (slot CC)
       (slot SVCCD)
       (slot PLNNOAC)
       (slot ACTYPE)                                        ; only one used
       (slot CALLSIGNNAME)
       (slot CALLSIGNID)
       (slot MSNTYPE)
       (slot APPROVED)
       (slot final-landing (default -1))
       (slot uuid)                                                 ; make sure
there can't be duplicates
)

(deftemplate AIRCRAFT
       (slot ACTYPE)
       (slot MAXSPEED)                                             ;miles per
hour
       (slot WEIGHT)                                        ;pounds
       (slot REQUIRED-RUNWAY-LENGTH)                  ;miles
       (slot RANGE)                                         ;miles
       (slot REQUIRED-OFFLOAD-TIME)                   ;minutes
)


(deftemplate AIRBASE
    (slot ID)
    (slot MAX-RUNWAY-LENGTH)                          ;miles
    (slot SUPPORTED-WEIGHT)                                 ;pounds
    (slot LATITUDE)
    (slot LONGITUDE)
    (slot LIGHTING-RESTRICTION-TIME-ON)        ;time (GMT)
    (slot LIGHTING-RESTRICTION-TIME-OFF)       ;time (GMT)
    (multislot EXCLUDED-ACTYPEIDS)
    (slot TURN-AROUND-TIME)                                 ; minutes
)

(deftemplate AIRSPACE
       (slot ID)
       (slot LATITUDE1)                                     ;top left
       (slot LONGITUDE1)
       (slot LATITUDE2)                                     ;bottom right
       (slot LONGITUDE2)
       (slot TYPE)
)

(deftemplate MISSION-OBJECTIVE
       (slot ACTYPE)
       (slot AIRBASE)
       (slot TIME)                                                 ;data time
       (slot earliest-time-reached (default -1))
)


;*******************************************************************************
*********************
;*** WELL-FORMEDNESS
;*** ---------------------------------------------------------------------------
---------------------
;*******************************************************************************
*********************
(defrule MAIN:mission-event-has-illegal-values
    (declare (salience 100))
       (MISSION_EVENT_ROW (EVENT_TYPE ?type) (EVENT_SEQ_ID ?id) (LOCATION_ID
?loc) (LATITUDE ?lat) (LONGITUDE ?long) (PLANNED_TIME_OFF ?off)
(PLANNED_TIME_ON ?on))
       (test (illformed-mission-event-row ?type ?id ?loc ?lat ?long ?off ?on))
       =>
       (halt))
(defrule MAIN:mission-object-has-illegal-values
    (declare (salience 100))
       (MISSIONOBJECT (ACTYPE ?actype) )
       (test (illformed-mission-object ?actype))
       =>
       (halt))
;(defrule MAIN:airspace-has-illegal-type
;   (declare (salience 100))
;      (AIRSPACE (ID ?id) (TYPE ?type&~"REFUEL"&~"RESTRICTED"))
;      =>
;      (error-feedback "airspace has illegal type " ?id " " ?type)(halt))
;(defrule MAIN:mission-event-has-illegal-event-type
;    (declare (salience 100))
;      (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE
?type&~"TO"&~"WAYPOINT"&~"REFUEL"&~"LDG"&~"OFFLOAD"))
;      =>
;      (error-feedback "mission-event-has-illegal-event-type " ?id " "
?type)(halt))
(defrule MAIN:mission-event-has-duplicate-sequence-id
    (declare (salience 80))
       ?f1 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) )
       ?f2 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id2&:(eq ?id1 ?id2)) )
       (test (not (eq ?f1 ?f2)))
       =>
       (error-feedback "mission-event-row-has-duplicate-sequence-id "
?id1)(halt))


;*****************************************************************************
*********************
;*** LINKED LIST GENERATOR
;*** -------------------------------------------------------------------------
---------------------
;*** PREV-LANDING generator
;*** PREV and NEXT generator
;*** DISTANCE-SINCE-LAST-"REFUEL" generator
;*****************************************************************************
*********************

(defrule MAIN:prev-and-next-generator
    (declare (salience 60))
       ?m1 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1))
       ?m2 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id2&:(< ?id1 ?id2)) (prev
?p2&:(< ?p2 ?id1)))
       =>
       (modify ?m2 (prev ?id1))
       (modify ?m1 (next ?id2)))

(defrule MAIN:prev-landing-generator
    (declare (salience 60))
       ?m1 <- (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id1))
       ?m2 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id2&:(< ?id1 ?id2)) (prev-
landing ?l&:(< ?l ?id1)))
       =>
       (modify ?m2 (prev-landing ?id1)))

(defrule MAIN:next-landing-generator
    (declare (salience 60))
       (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id1))
       ?m2 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id2&:(> ?id1 ?id2)) (next-
landing ?l&:(or (eq ?l -1) (> ?l ?id1))) )
       =>
       (modify ?m2 (next-landing ?id1)))

(defrule MAIN:final-landing-generator
    (declare (salience 60))
       (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id))
       ?o2 <- (MISSIONOBJECT (final-landing ?l&:(or (eq ?l -1) (< ?l ?id))) )
       =>
       (modify ?o2 (final-landing ?id)))

(defrule MAIN:distance-since-last-refuel-generator
    (declare (salience 40))
       ?m1 <- (MISSION_EVENT_ROW (EVENT_TYPE ?&"REFUEL"|"TO") (EVENT_SEQ_ID
?id1) (distance-since-last-refuel ?lr&:(= ?lr -1)) )
       =>
       (modify ?m1 (distance-since-last-refuel 0.0)))

(defrule MAIN:distance-since-last-refuel-generator-cont
    (declare (salience 40))
       ?m1 <- (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (LATITUDE ?lat1) (LONGITUDE
?long1) (distance-since-last-refuel ?lr1&:(> ?lr1 -1)) )
       ?m2 <- (MISSION_EVENT_ROW (EVENT_TYPE ?&~"REFUEL"&~"TO") (EVENT_SEQ_ID
?id2) (LATITUDE ?lat2) (LONGITUDE ?long2) (prev ?p&:(= ?p ?id1)) (distance-
since-last-refuel ?lr2&:(= ?lr2 -1)) )
```

```
	=>
	(modify ?m2 (distance-since-last-refuel (+ ?lr1 (lat-long-distance ?lat1
?long1 ?lat2 ?long2 "m")))))


;******************************************************************************
*********************
;*** OBJECTIVE RESTRICTION RULES
;*** --------------------------------------------------------------------------
---------------------
;*** Aircraft did not reach destination in specified time (it is allowed to
reach earlier)
;*** waypoint-refuel-or-landing-does-not-bring-aircraft-closer-to-destination
;******************************************************************************
*********************

(defrule MAIN:aircraft-did-not-reach-destination-in-specified-time-cont
	(MISSIONOBJECT (ACTYPE ?msnactype) (final-landing ?fl) )
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?&?fl) (EVENT_TYPE "LDG") (LOCATION_ID
?locid) (PLANNED_TIME_ON ?arrivedtime))
	(MISSION-OBJECTIVE (ACTYPE ?actype&:(eq ?actype ?msnactype)) (AIRBASE
?airbase) (TIME ?time))
	(test (or (neq ?airbase (location ?locid)) (> (date-time-in-minutes
?arrivedtime) (date-time-in-minutes ?time))))
	=>
	(error-feedback "aircraft-did-not-reach-destination-in-specified-time "
"aircraft: " ?actype " final destination " ?airbase " arrived at "
?arrivedtime))

(defrule MAIN:waypoint-refuel-or-landing-does-not-bring-aircraft-closer-to-
destination
	(MISSIONOBJECT (ACTYPE ?msnactype) (final-landing ?fl) )
	(MISSION_EVENT_ROW (EVENT_TYPE "WAYPOINT"|"REFUEL"|"TO") (EVENT_SEQ_ID
?id1) (LATITUDE ?lat1) (LONGITUDE ?long1))
	(MISSION_EVENT_ROW (EVENT_TYPE "WAYPOINT"|"REFUEL"|"LDG") (prev ?p2&:(=
?p2 ?id1))  (EVENT_SEQ_ID ?id2) (LATITUDE ?lat2) (LONGITUDE ?long2))
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?&?fl) (EVENT_TYPE "LDG") (LATITUDE
?lat3) (LONGITUDE ?long3))
	(test(< (lat-long-distance ?lat1 ?long1 ?lat3 ?long3 "m") (lat-long-
distance ?lat2 ?long2 ?lat3 ?long3 "m")))
	=>
	(error-feedback "waypoint-refuel-or-landing-does-not-bring-aircraft-
closer-to-destination - leg from: " ?id1 " to: " ?id2 " aircraft: "
?msnactype))

(defrule MAIN:destination-calculator
	(MISSIONOBJECT (ACTYPE ?msnactype) (final-landing ?fl) )
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (EVENT_TYPE ?t1) (LATITUDE ?lat1)
(LONGITUDE ?long1))
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id3&?fl) (EVENT_TYPE "LDG") (LATITUDE
?lat3) (LONGITUDE ?long3))
	=>
	(error-feedback "the distance between " ?id1 " ("?t1") and its final
landing " ?id3 " is " (lat-long-distance  ?lat1 ?long1 ?lat3 ?long3 "m") "
miles"))

;(defrule MAIN:aircraft-did-not-reach-destination-in-specified-time
;	(declare (salience 100))
;	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE "LDG") (LOCATION_ID
?locid) (PLANNED_TIME_OFF ?arrivedtime))
;	(MISSIONOBJECT (ACTYPE ?msnactype))
```

```
;      ?o <- (MISSION-OBJECTIVE (AIRBASE ?abid&:(eq ?abid (location ?locid)))
(ACTYPE ?actype&:(eq ?actype ?msnactype)) (earliest-time-reached ?earliesttime)
)
;      (test (or (eq ?earliesttime -1) (> (date-time-in-minutes ?earliesttime)
(date-time-in-minutes ?arrivedtime))))
;      =>
;      (modify ?o (earliest-time-reached ?arrivedtime)))
;(defrule MAIN:aircraft-did-not-reach-destination-in-specified-time-cont
;      (MISSIONOBJECT (ACTYPE ?msnactype))
;      (MISSION-OBJECTIVE (ACTYPE ?actype&:(eq ?actype ?msnactype)) (TIME ?time)
(earliest-time-reached ?earliesttime&:(or (eq ?earliesttime -1) (> (date-time-
in-minutes ?earliesttime) (date-time-in-minutes ?time)))))
;      =>
;      (error-feedback "aircraft-did-not-reach-destination-in-specified-time "
?actype))



;*******************************************************************************
********************
;*** AIRBASE RESTRICTION RULES
;*** ---------------------------------------------------------------------------
--------------------
;*** Aircraft cannot exceed runway length for takeoffs and landings
;*** Aircraft cannot takeoff or land at Airbase
;*** Aircraft cannot exceeed supported weight of airbase
;*** Aircraft cannot land or takeoff at night at this airbase
;*******************************************************************************
********************

(defrule MAIN:aircraft-cannot-exceed-runway-length-for-takeoffs-and-landings
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE "LDG"|"TO")
(LOCATION_ID ?locid) )
       (AIRBASE (ID ?abid&:(eq ?abid (location ?locid))) (MAX-RUNWAY-LENGTH
?abrwlen) )
       (MISSIONOBJECT (ACTYPE ?msnactype))
       (AIRCRAFT (ACTYPE ?actype&:(eq ?actype ?msnactype)) (REQUIRED-RUNWAY-
LENGTH ?acrwlen&:(> ?acrwlen ?abrwlen)) )
       =>
       (error-feedback "aircraft-cannot-exceed-runway-length-for-takeoffs-and-
landings " ?id " aircraft: " ?actype " airbase: " ?abid " actual runway length:
" ?abrwlen " but required length: " ?acrwlen))

(defrule MAIN:aircraft-cannot-takeoff-or-land-at-airbase
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE "LDG"|"TO")
(LOCATION_ID ?locid) )
       (AIRBASE (ID ?abid&:(eq ?abid (location ?locid))) (EXCLUDED-ACTYPEIDS
$?exclactype) )
       (MISSIONOBJECT (ACTYPE ?msnactype&:(> (length$ (intersection$ (create$
?msnactype) $?exclactype)) 0)))
       =>
       (error-feedback "aircraft-cannot-takeoff-or-land-at-airbase " ?id "
airbase: " ?locid " aircraft: " ?msnactype))

(defrule MAIN:aircraft-cannot-exceed-supported-weight-of-airbase
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE "LDG"|"TO")
(LOCATION_ID ?locid) )
       (AIRBASE (ID ?abid&:(eq ?abid (location ?locid))) (SUPPORTED-WEIGHT
?abweight) )
       (MISSIONOBJECT (ACTYPE ?msnactype))
       (AIRCRAFT (ACTYPE ?actype&:(eq ?actype ?msnactype)) (WEIGHT ?acweight&:(>
?acweight ?abweight)) )
       =>
```

```
	(error-feedback "aircraft-cannot-exceed-supported-weight-of-airbase " ?id
" aircraft: " ?actype " airbase: " ?abid " actual airbase weight: " ?abweight "
but required weight: " ?acweight))

(defrule MAIN:aircraft-cannot-land-or-takeoff-at-night-at-this-airbase
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (EVENT_TYPE "LDG"|"TO")
(LOCATION_ID ?locid) (PLANNED_TIME_ON ?ton))
	(AIRBASE (ID ?abid&:(eq ?abid (location ?locid))) (LIGHTING-RESTRICTION-
TIME-ON ?nolandon) (LIGHTING-RESTRICTION-TIME-OFF ?nolandoff) )
	(test (or
		(and (< (time-in-minutes ?nolandon) (time-in-minutes ?nolandoff))
(<= (time-in-minutes ?nolandon) (time-in-minutes ?ton)) (>= (time-in-minutes
?nolandoff) (time-in-minutes ?ton)))
		(and (> (time-in-minutes ?nolandon) (time-in-minutes ?nolandoff))
(not (and (<= (time-in-minutes ?nolandoff) (time-in-minutes ?ton)) (>= (time-
in-minutes ?nolandon) (time-in-minutes ?ton)))))
	))
	=>
	(error-feedback "aircraft-cannot-land-or-takeoff-at-night-at-this-airbase
" ?id " " ?locid " cannot land/takeoff at: " ?ton " " (time-in-minutes ?ton) ))


;******************************************************************************
*********************
;*** TIME RESTRICTION RULES
;*** --------------------------------------------------------------------------
---------------------
;*** Leg must start after the end of the immediately preceding Leg
;*** offload-and-refuel-must-have-end-time-after-the-start-time
;*** takeoff-landing-and-waypoint-must-have-end-time-equal-start-time
;*** Aircraft turnaround time exceeds airbase minimum turnaround time
;*** Aircraft offload time exceeds allocated time
;******************************************************************************
*********************

(defrule MAIN:leg-must-start-after-the-end-of-the-immediately-preceding-leg
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (PLANNED_TIME_OFF ?toff1))
	(MISSION_EVENT_ROW (EVENT_SEQ_ID ?id2) (prev ?p&:(= ?p ?id1))
(PLANNED_TIME_ON ?ton2&:(>= (date-time-in-minutes ?toff1) (date-time-in-minutes
?ton2))))
	=>
	(error-feedback "leg-must-start-after-the-end-of-the-immediately-
preceding-leg " ?id1 " " ?id2))

(defrule MAIN:offload-and-refuel-must-have-end-time-after-the-start-time
	(MISSION_EVENT_ROW (EVENT_TYPE "OFFLOAD"|"REFUEL") (EVENT_SEQ_ID ?id1)
(PLANNED_TIME_ON ?ton1) (PLANNED_TIME_OFF ?toff1&:(>= (date-time-in-minutes
?ton1) (date-time-in-minutes ?toff1))))
	=>
	(error-feedback "offload-and-refuel-must-have-end-time-after-the-start-
time " ?id1))

(defrule MAIN:takeoff-landing-and-waypoint-must-have-end-time-equal-start-time
	(MISSION_EVENT_ROW (EVENT_TYPE "TO"|"LDG"|"WAYPOINT") (EVENT_SEQ_ID ?id1)
(PLANNED_TIME_ON ?ton1) (PLANNED_TIME_OFF ?toff1&:(not (eq (date-time-in-
minutes ?ton1) (date-time-in-minutes ?toff1)))))
	=>
	(error-feedback "offload-and-refuel-must-end-after-the-start " ?id1))

(defrule MAIN:aircraft-turnaround-time-exceeds-airbase-minimum-turnaround-time
	(MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id1) (LOCATION_ID
?locid) (PLANNED_TIME_OFF ?toff1) )
```

```
       (MISSION_EVENT_ROW (EVENT_TYPE "TO") (EVENT_SEQ_ID ?id2) (PLANNED_TIME_ON
?ton2) (prev-landing ?l&:(eq ?id1 ?l)) )
       (MISSIONOBJECT (ACTYPE ?msnactype))
       (AIRBASE (ID ?abid&:(eq ?abid (location ?locid))) (TURN-AROUND-TIME
?turnaround&:(> ?turnaround (- (date-time-in-minutes ?ton2) (date-time-in-
minutes ?toff1)))) )
       =>
       (error-feedback "aircraft-turnaround-time-exceeds-airbase-minimum-
turnaround-time " ?id1 " " ?id2 " minimum: " ?turnaround " actual: " (- (date-
time-in-minutes ?ton2) (date-time-in-minutes ?toff1))))

(defrule MAIN:aircraft-offload-time-exceeds-allocated-time
       (MISSION_EVENT_ROW (EVENT_TYPE "OFFLOAD") (EVENT_SEQ_ID ?id1)
(LOCATION_ID ?locid) (PLANNED_TIME_ON ?ton1) (PLANNED_TIME_OFF ?toff1) )
       (MISSIONOBJECT (ACTYPE ?msnactype))
       (AIRCRAFT (ACTYPE ?actype&:(eq ?actype ?msnactype)) (REQUIRED-OFFLOAD-
TIME ?acoffloadtime&:(> ?acoffloadtime (- (date-time-in-minutes ?toff1) (date-
time-in-minutes ?ton1)))) )
       =>
       (error-feedback "aircraft-offload-time-exceeds-allocated-time " ?id1 "
needed: " ?acoffloadtime " allocated: " (- (date-time-in-minutes ?toff1) (date-
time-in-minutes ?ton1))))


;******************************************************************************
*********************
;*** SPACE RESTRICTION RULES
;*** --------------------------------------------------------------------------
---------------------
;*** "OFFLOAD" must occur at the same place as the LANDING that preceded it
(NOTE: close<100 miles)
;*** TAKEOFF must occur at the same place as the LANDING that preceded it
(NOTE: close<100 miles)
;*** Aircraft cannot exceed its maximum range
;*** Aircraft cannot travel the required distance in time
;*** Aircraft cannot refuel outside refuel airspace
;*** Aircraft cannot takeoff or land inside restricted airspace
;******************************************************************************
*********************

; we allow 100 miles because it is hard to click on the same place in MAF
(defrule MAIN:airbase-lat-long-does-not-match-the-mission-location-lat-long
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (LOCATION_ID ?locid1) (LATITUDE
?lat1) (LONGITUDE ?long1))
       (AIRBASE (ID ?airbase&:(eq ?airbase (location ?locid1))) (LATITUDE ?lat2)
(LONGITUDE ?long2))
       (test(> (lat-long-distance ?lat1 ?long1 ?lat2 ?long2 "m") 100))
       =>
       (error-feedback "airbase-lat-long-does-not-match-the-mission-location-
lat-long " ?id1 " " ?id2 " distance=" (lat-long-distance ?lat1 ?long1 ?lat2
?long2 "m")))

; we allow 100 miles because it is hard to click on the same place in MAF
(defrule MAIN:offload-must-occur-at-the-same-place-as-the-landing-that-
preceded-it
       (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id1) (LOCATION_ID
?locid1) (LATITUDE ?lat1) (LONGITUDE ?long1))
       (MISSION_EVENT_ROW (EVENT_TYPE "OFFLOAD") (EVENT_SEQ_ID ?id2) (prev
?p&:(= ?p ?id1)) (LOCATION_ID ?locid2) (LATITUDE ?lat2) (LONGITUDE ?long2))
       (test(or (neq (location ?locid1) (location ?locid2)) (> (lat-long-
distance ?lat1 ?long1 ?lat2 ?long2 "m") 100)))
       =>
       (error-feedback "offload-must-occur-at-the-same-place-as-the-landing-
that-preceded-it " ?id1 " " ?id2 " distance=" (lat-long-distance ?lat1 ?long1
?lat2 ?long2 "m")))

; we allow 100 miles because it is hard to click on the same place in MAF
(defrule MAIN:takeoff-must-occur-at-the-same-place-as-the-landing-that-
preceded-it
       (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id1) (LOCATION_ID
?locid1) (LATITUDE ?lat1) (LONGITUDE ?long1))
       (MISSION_EVENT_ROW (EVENT_TYPE "TO") (EVENT_SEQ_ID ?id2) (LOCATION_ID
?locid2) (prev-landing ?l&:(= ?l ?id1)) (LATITUDE ?lat2) (LONGITUDE ?long2))
       (test(or (neq (location ?locid1) (location ?locid2)) (> (lat-long-
distance ?lat1 ?long1 ?lat2 ?long2 "m") 100)))
       =>
       (error-feedback "takeoff-must-occur-at-the-same-place-as-the-landing-
that-preceded-it " ?id1 " " ?id2 " distance=" (lat-long-distance ?lat1 ?long1
?lat2 ?long2 "m")))

(defrule MAIN:aircraft-cannot-exceed-its-maximum-range
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id) (distance-since-last-refuel ?d) )
       (AIRCRAFT (ACTYPE ?actype) (RANGE ?r&:(> ?d ?r)))
       (MISSIONOBJECT (ACTYPE ?msnactype&:(eq ?actype ?msnactype)))
       =>
       (error-feedback "aircraft-cannot-exceed-its-maximum-range " ?id "
aircraft: " ?msnactype " distance: " ?d " but max range: " ?r))

;(defrule MAIN:waypoint-or-refuel-does-not-bring-aircraft-closer-to-destination
;      (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id3) (LATITUDE
?lat3) (LONGITUDE ?long3))
;      (MISSION_EVENT_ROW (EVENT_TYPE "WAYPOINT"|"REFUEL") (EVENT_SEQ_ID ?id2)
(LATITUDE ?lat2) (LONGITUDE ?long2) (next-landing ?nl2&:(eq ?nl2 ?id3)) )
;      (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (next ?n1&:(eq ?n1 ?id2))
(LATITUDE ?lat1) (LONGITUDE ?long1))
;      (test(< (lat-long-distance ?lat1 ?long1 ?lat3 ?long3 "m") (lat-long-
distance ?lat2 ?long2 ?lat3 ?long3 "m")))
;      =>
;      (error-feedback "waypoint-or-refuel-does-not-bring-aircraft-closer-to-
destination " ?id2 ))

(defrule MAIN:aircraft-cannot-travel-the-required-distance-in-time
       (MISSION_EVENT_ROW (EVENT_SEQ_ID ?id1) (LATITUDE ?lat1) (LONGITUDE
?long1) (PLANNED_TIME_OFF ?toff1) )
       (MISSION_EVENT_ROW (EVENT_TYPE "WAYPOINT"|"REFUEL"|"LDG") (EVENT_SEQ_ID
?id2) (LATITUDE ?lat2) (LONGITUDE ?long2) (PLANNED_TIME_ON ?ton2) (prev ?p&:(eq
?p ?id1)) )
       (MISSIONOBJECT (ACTYPE ?msnactype))
       (AIRCRAFT (ACTYPE ?actype&:(eq ?actype ?msnactype)) (MAXSPEED ?acspeed) )
       (test (< (* ?acspeed (/ (- (date-time-in-minutes ?ton2) (date-time-in-
minutes ?toff1)) 60)) (lat-long-distance ?lat1 ?long1 ?lat2 ?long2 "m")))
       =>
       (error-feedback "aircraft-cannot-travel-the-required-distance-in-time "
?id1 " " ?id2 " distance: " (lat-long-distance ?lat1 ?long1 ?lat2 ?long2 "m") "
time: " (- (date-time-in-minutes ?ton2) (date-time-in-minutes ?toff1)) "
max.speed: " ?acspeed))

; top > bottom and left<right
; TODO: there are no spaces that overflow on the right or left
(defrule MAIN:aircraft-cannot-refuel-outside-refuel-airspace
    (declare (salience 40))
       ?m <- (MISSION_EVENT_ROW (EVENT_TYPE "REFUEL") (EVENT_SEQ_ID ?id)
(LATITUDE ?lat) (LONGITUDE ?long) (inside-refueling-area "NO"))
       (AIRSPACE (TYPE "REFUEL") (LATITUDE1 ?lat1) (LONGITUDE1 ?long1)
(LATITUDE2 ?lat2) (LONGITUDE2 ?long2))
```

```
       (test (and (< ?long ?long2) (> ?long ?long1) (> ?lat ?lat2) (< ?lat
?lat1)))
       =>
       (modify ?m (inside-refueling-area "YES")))
(defrule MAIN:aircraft-cannot-refuel-outside-refuel-airspace-cont
       ?m <- (MISSION_EVENT_ROW (EVENT_TYPE "REFUEL") (EVENT_SEQ_ID ?id)
(LATITUDE ?lat) (LONGITUDE ?long) (inside-refueling-area "NO"))
       =>
       (error-feedback "aircraft-cannot-refuel-outside-refuel-airspace " ?id "
at lat: " ?lat " long: " ?long ))

; top > bottom and left<right
; TODO: there are no spaces that overflow on the right or left
(defrule MAIN:aircraft-cannot-takeoff-or-land-inside-restricted-airspace
       (MISSION_EVENT_ROW (EVENT_TYPE "TO"|"LDG") (EVENT_SEQ_ID ?id) (LATITUDE
?lat) (LONGITUDE ?long) )
       (AIRSPACE (TYPE "RESTRICTED") (LATITUDE1 ?lat1) (LONGITUDE1 ?long1)
(LATITUDE2 ?lat2) (LONGITUDE2 ?long2))
       (test (and (< ?long ?long2) (> ?long ?long1) (> ?lat ?lat2) (< ?lat
?lat1)))
       =>
       (error-feedback "aircraft-cannot-takeoff-or-land-inside-restricted-
airspace " ?id " at lat: " ?lat " long: " ?long))


;*****************************************************************************
*********************
;*** ORDERING RULES
;*** -------------------------------------------------------------------------
---------------------
;*** "OFFLOAD" must be immediately preceded by a LANDING
;*** "WAYPOINT" must be immediately preceded by a TAKEOFF, "WAYPOINT", or
"REFUEL"
;*** "REFUEL" must be immediately preceded by a TAKEOFF, "WAYPOINT", or
"REFUEL"
;*** LANDING must be immediately preceded by a TAKEOFF, "WAYPOINT", or "REFUEL"
;*** TAKEOFF must be immediately preceded by a LANDING or "OFFLOAD"
;*** First leg must be a TAKEOFF
;*** Last leg must be a LANDING or "OFFLOAD"
;*****************************************************************************
*********************

(defrule MAIN:offload-must-be-preceded-by-a-landing
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"LDG") (EVENT_SEQ_ID ?id1))
       (MISSION_EVENT_ROW (EVENT_TYPE "OFFLOAD") (EVENT_SEQ_ID ?id2) (prev
?p&:(= ?p ?id1)))
       =>
       (error-feedback "offload-must-be-preceded-by-a-landing " ?id1 " " ?id2 "
"))

(defrule MAIN:waypoint-must-be-immediately-preceded-by-a-takeoff-waypoint-or-
refuel
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"TO"&~"WAYPOINT"&~"REFUEL")
(EVENT_SEQ_ID ?id1))
       (MISSION_EVENT_ROW (EVENT_TYPE "WAYPOINT") (EVENT_SEQ_ID ?id2) (prev
?p&:(= ?p ?id1)))
       =>
       (error-feedback "waypoint-must-be-immediately-preceded-by-a-takeoff-
waypoint-or-refuel " ?id1 " " ?id2 " "))

(defrule MAIN:refuel-must-be-immediately-preceded-by-a-takeoff-waypoint-or-
refuel
```

```
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"TO"&~"WAYPOINT"&~"REFUEL")
(EVENT_SEQ_ID ?id1))
       (MISSION_EVENT_ROW (EVENT_TYPE "REFUEL") (EVENT_SEQ_ID ?id2) (prev ?p&:(=
?p ?id1)))
       =>
       (error-feedback "refuel-must-be-immediately-preceded-by-a-takeoff-
waypoint-or-refuel " ?id1 " " ?id2 " "))

(defrule MAIN:landing-must-be-immediately-preceded-by-a-takeoff-waypoint-or-
refuel
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"TO"&~"WAYPOINT"&~"REFUEL")
(EVENT_SEQ_ID ?id1))
       (MISSION_EVENT_ROW (EVENT_TYPE "LDG") (EVENT_SEQ_ID ?id2) (prev ?p&:(= ?p
?id1)))
       =>
       (error-feedback "landing-must-be-immediately-preceded-by-a-takeoff-
waypoint-or-refuel " ?id1 " " ?id2 " "))

(defrule MAIN:takeoff-must-be-immediately-preceded-by-a-landing-or-offload
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"LDG"&~"OFFLOAD") (EVENT_SEQ_ID ?id1))
       (MISSION_EVENT_ROW (EVENT_TYPE "TO") (EVENT_SEQ_ID ?id2) (prev ?p&:(= ?p
?id1)))
       =>
       (error-feedback "takeoff-must-be-immediately-preceded-by-a-landing-or-
offload " ?id1 " " ?id2 " "))

(defrule MAIN:first-leg-must-be-a-takeoff
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"TO") (EVENT_SEQ_ID ?id1) (prev -1))
       =>
       (error-feedback "first-leg-must-be-a-takeoff " ?id1))

(defrule MAIN:last-leg-must-be-a-landing-or-offload
       (MISSION_EVENT_ROW (EVENT_TYPE ?&~"LDG"&~"OFFLOAD") (EVENT_SEQ_ID ?id1)
(next -1))
       =>
       (error-feedback "last-leg-must-be-a-landing-or-offload " ?id1))
```

APPENDIX B: Initial Data

Mission Objectives

```
(deffacts facts "mission objectives"
(MISSION-OBJECTIVE (ACTYPE "C17") (AIRBASE "KFSM") (TIME "20051020T130000Z"))
(MISSION-OBJECTIVE (ACTYPE "RF5A") (AIRBASE "GOOY") (TIME "20051020T230000Z"))
(MISSION-OBJECTIVE (ACTYPE "B52G") (AIRBASE "LTAG") (TIME "20051020T180000Z"))
(MISSION-OBJECTIVE (ACTYPE "F111D") (AIRBASE "KGVW") (TIME "20051020T150000Z"))
)
```

Airspaces

```
(deffacts facts "airspaces"
(AIRSPACE (LATITUDE1 26) (LONGITUDE1 45) (LATITUDE2 20) (LONGITUDE2 50) (TYPE
"RESTRICTED"))
(AIRSPACE (LATITUDE1 30) (LONGITUDE1 -105) (LATITUDE2 25) (LONGITUDE2 -100)
(TYPE "REFUEL"))
(AIRSPACE (LATITUDE1 31) (LONGITUDE1 25) (LATITUDE2 15) (LONGITUDE2 33) (TYPE
"RESTRICTED"))
(AIRSPACE (LATITUDE1 40) (LONGITUDE1 -30) (LATITUDE2 35) (LONGITUDE2 -25) (TYPE
"REFUEL"))
(AIRSPACE (LATITUDE1 45) (LONGITUDE1 0) (LATITUDE2 40) (LONGITUDE2 5) (TYPE
"RESTRICTED"))
(AIRSPACE (LATITUDE1 55) (LONGITUDE1 -5) (LATITUDE2 50) (LONGITUDE2 0) (TYPE
"REFUEL"))
(AIRSPACE (LATITUDE1 65) (LONGITUDE1 -25) (LATITUDE2 60) (LONGITUDE2 -20) (TYPE
"REFUEL"))
)
```

Airbases

```
(deffacts facts "airbases"
(AIRBASE (ID "^001") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "31") (LONGITUDE "45") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "^002") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "29") (LONGITUDE "46") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "^003") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "30") (LONGITUDE "40") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "^004") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "30") (LONGITUDE "48") (TURN-AROUND-TIME "60")(LIGHTING-RESTRICTION-
TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF "090000Z"))
(AIRBASE (ID "^005") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "31") (LONGITUDE "44") (TURN-AROUND-TIME "60")(EXCLUDED-ACTYPEIDS
"E3A" "T38A" ))
(AIRBASE (ID "AABB") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "29") (LONGITUDE "45.25") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "BIKF") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "63.9861111111111") (LONGITUDE "-22.6083333333333") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "KC130" "E3" ))
(AIRBASE (ID "CG01") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "42.8958333333333") (LONGITUDE "15.8555555555556") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "CG02") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "42.0922222222222") (LONGITUDE "17.5405555555556") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "A10A" ))
(AIRBASE (ID "CG03") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "39.3888888888889") (LONGITUDE "17.8008333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "CV01") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "28") (LONGITUDE "49") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "CV49") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "36") (LONGITUDE "126") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "CV54") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "36") (LONGITUDE "130") (TURN-AROUND-TIME "120")(LIGHTING-
RESTRICTION-TIME-ON "050000Z") (LIGHTING-RESTRICTION-TIME-OFF "150000Z"))
(AIRBASE (ID "CV69") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "29") (LONGITUDE "50") (TURN-AROUND-TIME "60")(EXCLUDED-ACTYPEIDS
"S3A" "HC130N" "A10" ))
(AIRBASE (ID "CV99") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "28") (LONGITUDE "50") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "CVBG") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "29") (LONGITUDE "50") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "CYQB") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "47.5063888888889") (LONGITUDE "8.59916666666667") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "CYYC") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "47.4377777777778") (LONGITUDE "8.68777777777778") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "DPG") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.2") (LONGITUDE "-112.936666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "EDAF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "50.0319444444444") (LONGITUDE "8.56944444444444") (TURN-AROUND-TIME
"60"))
```

```
(AIRBASE (ID "EDAR") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "49.4361111111111") (LONGITUDE "7.6") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "EGUL") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "52.3666666666667") (LONGITUDE "0.502777777777778") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "EGUN") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "52.36") (LONGITUDE "0.486666666666667") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "F14" "ES3A" "A10A" ))
(AIRBASE (ID "EGVA") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "51.0669444444444") (LONGITUDE "-1.06861111111111") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "EGWZ") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "52.375") (LONGITUDE "0.219444444444444") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "ENBO") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "67.0183333333333") (LONGITUDE "14.0338888888889") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "ETAD") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40") (LONGITUDE "50") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "ETAR") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "49.4333333333333") (LONGITUDE "7.6") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "FJDG") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "-7.3") (LONGITUDE "72.4") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "FOK") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.8433333333333") (LONGITUDE "-72.6316666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "GAV1") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "35.6983333333333") (LONGITUDE "126.961666666667") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "TORNDO" "AC130H" "TR1" ))
(AIRBASE (ID "GAV2") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "37.415") (LONGITUDE "128.451388888889") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "GOOY") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "14.7472222222222") (LONGITUDE "-17.4944444444444") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "HECW") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "30.0019444444444") (LONGITUDE "30.0847222222222") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KACY") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.4583333333333") (LONGITUDE "-74.5766666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KADW") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "38.8166666666667") (LONGITUDE "-76.8666666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KBAB") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.1333333333333") (LONGITUDE "-121.433333333333") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "KBAD") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "32.5") (LONGITUDE "-93.6666666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KBAF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "42.1583333333333") (LONGITUDE "-72.715") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KBDL") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "41.9383333333333") (LONGITUDE "-72.6833333333333") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "RC135V" "KC135A" ))
(AIRBASE (ID "KBEL") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "44.1461111111111") (LONGITUDE "75.6438888888889") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KBIX") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "30.4166666666667") (LONGITUDE "-88.9166666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KBKF") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "39.0461111111111") (LONGITUDE "-104.751666666667") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "130000Z") (LIGHTING-RESTRICTION-TIME-OFF
"230000Z"))
(AIRBASE (ID "KBLV") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "38.55") (LONGITUDE "-89.85") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KBOI") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "43.565") (LONGITUDE "-116.225") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KBTV") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "44.4733333333333") (LONGITUDE "-73.15") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KCBM") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "33.65") (LONGITUDE "-88.45") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KCHS") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.9") (LONGITUDE "-80.0333333333333") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KCOS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "38.9527777777778") (LONGITUDE "-104.633333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KCVS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "34.3833333333333") (LONGITUDE "-103.316666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KCYS") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "41.1558333333333") (LONGITUDE "-104.811944444444") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "130000Z") (LIGHTING-RESTRICTION-TIME-OFF
"230000Z"))
(AIRBASE (ID "KDFW") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.9") (LONGITUDE "-97.05") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KDMA") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.1666666666667") (LONGITUDE "-110.883333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KDOV") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.1333333333333") (LONGITUDE "-75.4666666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KDYS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "32.4166666666667") (LONGITUDE "-99.85") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "EC130" ))
(AIRBASE (ID "KEDW") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "34.0841666666667") (LONGITUDE "-117.084166666667") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "KEND") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "36.3333333333333") (LONGITUDE "-97.9166666666667") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KFFO") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.8333333333333") (LONGITUDE "-84.0583333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KFMH") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "41.6583333333333") (LONGITUDE "-70.5216666666667") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "F15C" ))
(AIRBASE (ID "KFOK") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "40.8433333333333") (LONGITUDE "-72.6316666666667") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z"))
(AIRBASE (ID "KFSM") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "35.3388888888889") (LONGITUDE "-94.3666666666667") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "A4D" "C130B" "C17" ))
(AIRBASE (ID "KGGG") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "32.3833333333333") (LONGITUDE "-94.7166666666667") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KGSB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "35.3333333333333") (LONGITUDE "-77.9666666666667") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z"))
(AIRBASE (ID "KGTB") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "44.05") (LONGITUDE "-75.6666666666667") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z")(EXCLUDED-ACTYPEIDS "F15E" "ES3A" "F14" ))
(AIRBASE (ID "KGTF") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "47.4819444444444") (LONGITUDE "-111.370555555556") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KGVW") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "38.85") (LONGITUDE "-94.5666666666667") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "140000Z") (LIGHTING-RESTRICTION-TIME-OFF
"000000Z"))
(AIRBASE (ID "KHIF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "41.1266666666667") (LONGITUDE "-111.971666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KHMN") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "32.85") (LONGITUDE "-106.1") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KHST") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "25.4833333333333") (LONGITUDE "-80.3833333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KIAB") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "37.6166666666667") (LONGITUDE "-97.2666666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KIAG") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "43.1066666666667") (LONGITUDE "-78.945") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "C130E" "F15C" ))
(AIRBASE (ID "KIKR") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "35.05") (LONGITUDE "-106.616666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KJAN") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "32.3166666666667") (LONGITUDE "-90.0833333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KLCH") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "30.1333333333333") (LONGITUDE "-93.2166666666667") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "140000Z") (LIGHTING-RESTRICTION-TIME-OFF
"000000Z"))
(AIRBASE (ID "KLFI") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37") (LONGITUDE "-76.0005555555556") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KLMT") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "42.1566666666667") (LONGITUDE "-121.733333333333") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "KLRF") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "34.9166666666667") (LONGITUDE "-92.15") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KLSV") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "0") (LONGITUDE "0") (TURN-AROUND-TIME "120")(LIGHTING-RESTRICTION-
TIME-ON "200000Z") (LIGHTING-RESTRICTION-TIME-OFF "060000Z"))
(AIRBASE (ID "KLTS") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "0") (LONGITUDE "0") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KMCF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "27.85") (LONGITUDE "-82.525") (TURN-AROUND-TIME "180")(LIGHTING-
RESTRICTION-TIME-ON "140000Z") (LIGHTING-RESTRICTION-TIME-OFF "000000Z"))
(AIRBASE (ID "KMEI") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "32.3333333333333") (LONGITUDE "-88.75") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KMIB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "48.4166666666667") (LONGITUDE "-101.35") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KMIJ") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "40.2") (LONGITUDE "-112.936666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KMLU") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "32.5166666666667") (LONGITUDE "-92.0333333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KMRB") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.4019444444444") (LONGITUDE "-77.9844444444444") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z")(EXCLUDED-ACTYPEIDS "F111A" "EF111A" "ES3A" ))
(AIRBASE (ID "KMTN") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "39.325") (LONGITUDE "-76.4133333333333") (TURN-AROUND-TIME "180"))
```

```
(AIRBASE (ID "KMUO") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "43.05") (LONGITUDE "-115.866666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KMXF") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.3833333333333") (LONGITUDE "-86.35") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KNBG") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "29.8333333333333") (LONGITUDE "-90.0333333333333") (TURN-AROUND-TIME
"120")(EXCLUDED-ACTYPEIDS "HC130N" "A7D" "FA18C" ))
(AIRBASE (ID "KNFL") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "39.4") (LONGITUDE "-118.701111111111") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "KNFW") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.7666666666667") (LONGITUDE "-97.45") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "F15E" ))
(AIRBASE (ID "KNKX") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.8683333333333") (LONGITUDE "-117.143333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KNMM") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.55") (LONGITUDE "-88.55") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KNQX") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "24.5833333333333") (LONGITUDE "-81.7166666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KNTD") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "34.1205555555556") (LONGITUDE "-119.126666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KNXX") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.2") (LONGITUDE "-75.15") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KOFF") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "41.1166666666667") (LONGITUDE "-95.9169444444444") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "GR1" "GR1" "B52G" ))
(AIRBASE (ID "KOQU") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "41.5972222222222") (LONGITUDE "-71.4122222222222") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KPAM") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "30.0722222222222") (LONGITUDE "-85.5833333333333") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "UH60" "C17" "EC130" ))
(AIRBASE (ID "KPBG") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "44.65") (LONGITUDE "-73.4666666666667") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "B52H" "A4D" ))
(AIRBASE (ID "KPDX") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "45.5875") (LONGITUDE "-122.598333333333") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KPIT") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "40.4916666666667") (LONGITUDE "-80.2333333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "KPOB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "35.1722222222222") (LONGITUDE "-79.025") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "KQUO") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "41.5972222222222") (LONGITUDE "-71.4122222222222") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KRCA") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "44.15") (LONGITUDE "-103.1") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KRDR") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "47.9666666666667") (LONGITUDE "-97.4") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KRIV") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "33.8805555555556") (LONGITUDE "-117.259444444444") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KRME") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "43.2336111111111") (LONGITUDE "-75.4069444444444") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z"))
(AIRBASE (ID "KRND") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "29.5388888888889") (LONGITUDE "-98.2861111111111") (TURN-AROUND-TIME
"120"))
```

```
(AIRBASE (ID "KRNO") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.4983333333333") (LONGITUDE "-119.768333333333") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z")(EXCLUDED-ACTYPEIDS "FA18D" ))
(AIRBASE (ID "KROW") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "33.3") (LONGITUDE "-104.533333333333") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "C141" "E8" ))
(AIRBASE (ID "KSCH") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "42.8566666666667") (LONGITUDE "-73.93") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z")(EXCLUDED-ACTYPEIDS "FA18C" ))
(AIRBASE (ID "KSHV") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.45") (LONGITUDE "-93.8333333333333") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KSKA") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "47.6166666666667") (LONGITUDE "-117.65") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KSKF") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "29.3666666666667") (LONGITUDE "-98.5833333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KSLC") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "40.7883333333333") (LONGITUDE "-111.978333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KSPS") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "33.9861111111111") (LONGITUDE "-98.4972222222222") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KSSC") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "33.9777777777778") (LONGITUDE "-80.4777777777778") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KSUU") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "38.2666666666667") (LONGITUDE "-121.933333333333") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "KSWF") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "41.5033333333333") (LONGITUDE "-74.105") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "150000Z") (LIGHTING-RESTRICTION-TIME-OFF
"010000Z"))
(AIRBASE (ID "KSYR") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "43.1116666666667") (LONGITUDE "-76.1066666666667") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "KSZL") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "38.7333333333333") (LONGITUDE "-93.55") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "140000Z") (LIGHTING-RESTRICTION-TIME-OFF
"000000Z"))
(AIRBASE (ID "KTIK") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "35.4166666666667") (LONGITUDE "-97.3833333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KTUL") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "36.1833333333333") (LONGITUDE "-95.8833333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KTUS") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "32.1163888888889") (LONGITUDE "-110.941388888889") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "130000Z") (LIGHTING-RESTRICTION-TIME-OFF
"230000Z")(EXCLUDED-ACTYPEIDS "RF4C" "KC10A" ))
(AIRBASE (ID "KU4Z") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "40.62") (LONGITUDE "-111.993333333333") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "KVPS") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "30.4888888888889") (LONGITUDE "-86.5333333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "KWRB") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "32.6333333333333") (LONGITUDE "-83.6") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "KWRI") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.0166666666667") (LONGITUDE "-74.6") (TURN-AROUND-TIME "120"))
```

```
(AIRBASE (ID "KYUM") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "32.6613888888889") (LONGITUDE "-114.358888888889") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "LEMO") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "37.0666666666667") (LONGITUDE "-5.61666666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "LERT") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "36.65") (LONGITUDE "-6.35") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "LETO") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.4833333333333") (LONGITUDE "-3.46666666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "LEZG") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "41.6666666666667") (LONGITUDE "-1.05") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "LEZL") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.4166666666667") (LONGITUDE "-5.9") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "LFMI") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "43.5283333333333") (LONGITUDE "4.92666666666667") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "S3B" "F14A" "C5A" ))
(AIRBASE (ID "LIBR") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "40.6666666666667") (LONGITUDE "17.95") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "LIBV") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "40.7666666666667") (LONGITUDE "16.9333333333333") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LICT") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "37.9166666666667") (LONGITUDE "12.4833333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "LICZ") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.4") (LONGITUDE "14.9166666666667") (TURN-AROUND-TIME
"180")(EXCLUDED-ACTYPEIDS "KC10A" "F14B" "E3B" ))
(AIRBASE (ID "LIED") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "39.35") (LONGITUDE "8.96666666666667") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "LIPA") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "46.0333333333333") (LONGITUDE "12.6") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LIPC") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "9000")
(LATITUDE "44.2333333333333") (LONGITUDE "12.3166666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "LIPI") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "45.9833333333333") (LONGITUDE "13.05") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "LIPL") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "45.4333333333333") (LONGITUDE "10.275") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LIPR") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "44.0238888888889") (LONGITUDE "12.6111111111111") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "LIPS") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "45.6888888888889") (LONGITUDE "12.0944444444444") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "LIPT") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "45.575") (LONGITUDE "11.5269444444444") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "LIPX") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "45.4") (LONGITUDE "10.8833333333333") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LIPY") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "43.6166666666667") (LONGITUDE "13.3666666666667") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "LIRN") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.8833333333333") (LONGITUDE "14.2833333333333") (TURN-AROUND-TIME
"180"))
```

```
(AIRBASE (ID "LIRP") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "43.6833333333333") (LONGITUDE "10.4") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LIRS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "42.75") (LONGITUDE "11.0666666666667") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "LIYW") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "46.0308333333333") (LONGITUDE "12.0516666666667") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "210000Z") (LIGHTING-RESTRICTION-TIME-OFF
"070000Z"))
(AIRBASE (ID "LPLA") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "38.775") (LONGITUDE "-27.1055555555556") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "LTAG") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "9000")
(LATITUDE "36.9966666666667") (LONGITUDE "32.425") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "220000Z") (LIGHTING-RESTRICTION-TIME-OFF
"080000Z"))
(AIRBASE (ID "NCON") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "-28.5") (LONGITUDE "-50.0027777777778") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "NID") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "35.6883333333333") (LONGITUDE "-117.69") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "120000Z") (LIGHTING-RESTRICTION-TIME-OFF
"220000Z"))
(AIRBASE (ID "NIKE") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "-27.1863888888889") (LONGITUDE "152") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "NUSA") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "27.1863888888889") (LONGITUDE "35.3727777777778") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "220000Z") (LIGHTING-RESTRICTION-TIME-OFF
"080000Z"))
(AIRBASE (ID "NUW") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "48.3519444444444") (LONGITUDE "-122.655555555556") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "FA18D" "TORNDO" ))
(AIRBASE (ID "OBBI") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "26.2666666666667") (LONGITUDE "50.6333333333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "OBBS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "25.9166666666667") (LONGITUDE "50.5833333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "OBE1") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "30") (LONGITUDE "43") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "OEAW") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "12.1863888888889") (LONGITUDE "133.558888888889") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "050000Z") (LIGHTING-RESTRICTION-TIME-OFF
"150000Z"))
(AIRBASE (ID "OEDF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "26.0005555555556") (LONGITUDE "49.0011111111111") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "OEDR") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "26.2666666666667") (LONGITUDE "50.15") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "OEHF") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "25") (LONGITUDE "49.0033333333333") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "OEJB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "11") (LONGITUDE "111") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "OEJD") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "21.5") (LONGITUDE "39.2") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "OEJN") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "21.6833333333333") (LONGITUDE "39.16") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "OEKH") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "24.05") (LONGITUDE "47.5666666666667") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "OEKK") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "27.8966666666667") (LONGITUDE "45.5266666666667") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "RF4B" "E3A" ))
(AIRBASE (ID "OEKM") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "18.3066666666667") (LONGITUDE "42.8116666666667") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF
"090000Z"))
(AIRBASE (ID "OEKW") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "29.3666666666667") (LONGITUDE "47.5166666666667") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "OEPA") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "28.0333333333333") (LONGITUDE "46.0022222222222") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF
"090000Z"))
(AIRBASE (ID "OERK") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "24.9616666666667") (LONGITUDE "46.7083333333333") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "OERY") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "24.7166666666667") (LONGITUDE "46.7333333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "OESA") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "27.0858333333333") (LONGITUDE "48.0683333333333") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "OETB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "28.3666666666667") (LONGITUDE "36.6333333333333") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "220000Z") (LIGHTING-RESTRICTION-TIME-OFF
"080000Z"))
(AIRBASE (ID "OETF") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "21.0355555555556") (LONGITUDE "40.0505555555556") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF
"090000Z")(EXCLUDED-ACTYPEIDS "C130E" "ES3A" ))
(AIRBASE (ID "OIKB") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "7000")
(LATITUDE "27.2166666666667") (LONGITUDE "56.3666666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "OKAF") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "29.2166666666667") (LONGITUDE "47.9666666666667") (TURN-AROUND-TIME
"180")(LIGHTING-RESTRICTION-TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF
"090000Z"))
(AIRBASE (ID "OKAJ") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "28.9333333333333") (LONGITUDE "47.8") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "OMAA") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "9000")
(LATITUDE "24.4388888888889") (LONGITUDE "54.65") (TURN-AROUND-TIME "60"))
(AIRBASE (ID "OSAW") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "27") (LONGITUDE "46") (TURN-AROUND-TIME "180")(LIGHTING-RESTRICTION-
TIME-ON "230000Z") (LIGHTING-RESTRICTION-TIME-OFF "090000Z"))
(AIRBASE (ID "OTBD") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "-33.5591666666667") (LONGITUDE "-111.559166666667") (TURN-AROUND-
TIME "180"))
(AIRBASE (ID "PAED") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "61.2555555555556") (LONGITUDE "-149.797222222222") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "PAEI") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "64.0525") (LONGITUDE "-147.001388888889") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "F111A" "S3B" "K1" ))
(AIRBASE (ID "PGUA") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "5000")
(LATITUDE "13.0513888888889") (LONGITUDE "144.085") (TURN-AROUND-TIME "120"))
(AIRBASE (ID "RJSM") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "40.7061111111111") (LONGITUDE "141.365") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "RJTY") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "9000")
(LATITUDE "35.7583333333333") (LONGITUDE "139.358333333333") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "050000Z") (LIGHTING-RESTRICTION-TIME-OFF
"150000Z"))
(AIRBASE (ID "RKJJ") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "35.1372222222222") (LONGITUDE "126.816666666667") (TURN-AROUND-TIME
"180"))
(AIRBASE (ID "RKJK") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "35.8919444444444") (LONGITUDE "126.618611111111") (TURN-AROUND-TIME
"120"))
```

```
(AIRBASE (ID "RKNN") (MAX-RUNWAY-LENGTH "1000") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.7563888888889") (LONGITUDE "128.958055555556") (TURN-AROUND-TIME
"120")(LIGHTING-RESTRICTION-TIME-ON "050000Z") (LIGHTING-RESTRICTION-TIME-OFF
"150000Z"))
(AIRBASE (ID "RKPP") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "35.1641666666667") (LONGITUDE "129.132222222222") (TURN-AROUND-TIME
"60")(LIGHTING-RESTRICTION-TIME-ON "050000Z") (LIGHTING-RESTRICTION-TIME-OFF
"150000Z")(EXCLUDED-ACTYPEIDS "AC130U" "OV10" ))
(AIRBASE (ID "RKSM") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.4594444444444") (LONGITUDE "127.115") (TURN-AROUND-TIME "180"))
(AIRBASE (ID "RKSO") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.0888888888889") (LONGITUDE "127.046944444444") (TURN-AROUND-TIME
"60"))
(AIRBASE (ID "RKSS") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "37.5438888888889") (LONGITUDE "127.173611111111") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "RKSW") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "7000")
(LATITUDE "37.2294444444444") (LONGITUDE "127.015555555556") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "RKTN") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "5000")
(LATITUDE "35.8866666666667") (LONGITUDE "128.670833333333") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "RKTY") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "36.6327777777778") (LONGITUDE "128.370277777778") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "RODN") (MAX-RUNWAY-LENGTH "400") (SUPPORTED-WEIGHT "7000")
(LATITUDE "26.3563888888889") (LONGITUDE "127.763888888889") (TURN-AROUND-TIME
"120"))
(AIRBASE (ID "RPVM") (MAX-RUNWAY-LENGTH "800") (SUPPORTED-WEIGHT "5000")
(LATITUDE "10.3111111111111") (LONGITUDE "129.980555555556") (TURN-AROUND-TIME
"60")(EXCLUDED-ACTYPEIDS "A7D" "OA4M" "E3A" ))
(AIRBASE (ID "XAMC") (MAX-RUNWAY-LENGTH "600") (SUPPORTED-WEIGHT "5000")
(LATITUDE "28") (LONGITUDE "45") (TURN-AROUND-TIME "60"))
)
```

Aircraft

```
(deffacts facts "aircrafts"
(AIRCRAFT (ACTYPE "A10") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "A10A") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "A4") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "A4D") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "A4M") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "A6E") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "A7D") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "AC130A") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "AC130H") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "AC130U") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "AH64") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "AH64A") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
```

```
(AIRCRAFT (ACTYPE "AV8") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "AV8B") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "B1B") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "B52G") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "B52H") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "C130A") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "C130B") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C130E") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "C130H") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "C141") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C141B") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "C160") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C17") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C23A") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "C5A") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "C5B") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C9") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "C9A") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "CH46NA") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "CH53") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "E2C") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "E3") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "E3A") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "E3B") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "E8") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "E8A") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "EA6B") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "EC130") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "EC130E") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "EC130H") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "EF111A") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
```

```
(AIRCRAFT (ACTYPE "ES3A") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F111A") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F111D") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F111F") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F117") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F14") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F14A") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F14B") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F14D") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F15A") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F15C") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F15D") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F15E") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F16") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F16A") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F16C") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "F16CJ") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F16D") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F4D") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F4E") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F4G") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "F5") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "F5E") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "FA18") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "FA18C") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "FA18D") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "GHAWK") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "GR1") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "HC130") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "HC130N") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "HC130P") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
```

```
(AIRCRAFT (ACTYPE "HH53") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "HH53E") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "HH58C") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "HH60") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "K1") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KA6") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KC10") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KC10A") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KC130") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "KC135") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "KC135A") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KC135E") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "KC135Q") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "KC135R") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "LC130") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "M2000") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "M2000R") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "OA10") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "OA10A") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "OA4M") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "OV10") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "OV10D") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "RC135V") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "RF4B") (MAXSPEED "1200") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "RF4C") (MAXSPEED "800") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "RF5A") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "S3A") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "120"))
(AIRCRAFT (ACTYPE "S3B") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "5000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "SH3") (MAXSPEED "800") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "SH60") (MAXSPEED "1200") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "4000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "SH60F") (MAXSPEED "1000") (WEIGHT "8000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "180"))
```

```
(AIRCRAFT (ACTYPE "T38A") (MAXSPEED "800") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "700") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "TORNDO") (MAXSPEED "1200") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "300") (RANGE "3000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "TR1") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "180"))
(AIRCRAFT (ACTYPE "U2") (MAXSPEED "1000") (WEIGHT "6000") (REQUIRED-RUNWAY-
LENGTH "900") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
(AIRCRAFT (ACTYPE "UH60") (MAXSPEED "1000") (WEIGHT "4000") (REQUIRED-RUNWAY-
LENGTH "500") (RANGE "6000") (REQUIRED-OFFLOAD-TIME "60"))
)
```